

University of Tartu
Faculty of Science and Technology
Institute of Mathematics and Statistics

Karen Danielyan

Swaption pricing with SABR model

Actuarial and Financial Engineering
Master's thesis (30 ECTS)

Supervisor: Raul Kangro

Tartu 2020

Swaption pricing with SABR model

Master's thesis

Karen Danielyan

Abstract. The purpose of this Master's thesis is to present the SABR and shifted SABR models and perform a calibration procedure on the Euribor and Libor swaptions' volatility cube. The first chapter focuses on the mathematical preliminaries for interest rate derivatives and money market. The second chapter tackles the classical option pricing models, in particular Black-76 model. The third chapter introduces SABR model and Hagan's approximation formulas, as well as the shifted SABR and the effects of SABR parameters on the volatility smiles and skews. Finally, the last chapter shows the detailed procedure of calibration of the Euribor and Libor volatility cubes, validation of the fit and some market data engineering examples referencing to the python code in the appendix.

CERCS research specialisation: P160 Statistics, operations research, programming, actuarial mathematics.

Keywords: SABR, Python, swaption volatility cube, volatility cube calibration, volatility surface, volatility smile, volatility skew, negative rates, Libor, Euribor, quote engineering.

Intressimäärade vahetusoptsioonide hinnastamine SABR mudeliga

Magistritöö

Karen Danielyan

Lühikokkuvõte. Selle magistritöö eesmärk on anda ülevaade SABR ja nihutatud SABR mudelist ning kalibreerida Euribori ja Libori svaptsioonide ehk intressimäära vahetusoptsioonide volatiilsusi. Töö esimeses peatükis antakse ülevaade intressimäära derivatiivide ja rahaturuga seotud baasmõistetest. Teine peatükk keskendub klassikaliste optsioonide hinnastamise mudelitele, täpsemalt Black-76 mudelile. Kolmandas peatükis tutvustatakse SABR mudelit, Hagani lähendi valemeid ja nihutatud SABR mudelit ning seletatakse SABRi parameetrite mõju volatiilsuskõveratele. Viimases peatükis tuuakse Pythoni koodi abil detailseid näiteid Euribori ja Libori volatiilsuste kalibreerimisprotseduurist, mudeli sobivuse valideerimisest ning turuandmete modelleerimisest.

CERCS teaduseriala: P160 Statistika, operatsioonianalüüs, programmeerimine, finants- ja kindlustusmatemaatika.

Märksõnad: SABR, Python (programmeerimiskeel), svaptsioonide volatiilsus, volatiilsuste kalibreerimine, volatiilsuspind, volatiilsuskõver, volatiilsusnihe, negatiivne intressimäär, Libor, Euribor, hinnapakkumise modelleerimine.

Acknowledgements

First of all, I want to express my gratitude and heartfelt thanks to my computational finance course teacher and my thesis supervisor Raul Kangro for sharing his knowledge and support throughout the writing process of my thesis.

I also want to thank Andrej Bajic for supervising my internship program in Deloitte Audit Analytics and for suggesting the topic for my master thesis. He provided me with insightful content and constructive feedback on my thesis.

Moreover, I want to express my appreciation to my Deloitte Audit Analytics colleagues for providing me with tools necessary for the practical application of the ideas presented in my thesis, particularly, Sabri Dali who provided a great assistance in my undertaking to tackle this topic.

Contents

Introduction	6
1 Mathematical Preliminaries	7
1.1 Money-market Preliminaries	7
1.2 Swaptions	13
2 Black 76 model	16
2.1 Black Scholes option pricing model	16
2.2 Bachelier’s model	19
2.3 Risk-neutral valuation	20
2.4 Black 76 model for swaptions	22
3 SABR model	26
3.1 SABR model	27
3.2 Breaking down SABR parameters	29
3.3 Shifted SABR model	32
4 Results	35
4.1 Market Data: Volatility Cube	35
4.2 Euribor volatility cube calibration with SABR	37
4.3 USD Libor volatility cube calibration with SABR	44
Conclusion	50
Appendix	52

Introduction

Interest rate swaptions are often priced using Black 76 model. In Black 76 model there is one-to-one relation between the price of the swaption and the volatility parameter σ_B , thus the option prices are often quoted by stating the implied volatility, i.e. the unique value of the volatility parameter, which yields the swaption's price when using Black 76 model. In practice, swaptions with different strikes require different volatilities to match their market prices resulting the volatility skews and smiles. Handling these skews and smiles are critical component of volatility modeling. One of the tools to address these requirements is the SABR (Stochastic α, β, ρ) model. Thus, in this thesis we will cover the SABR model and its practical application on Euribor and USD Libor swaptions, as well as the model's calibration on the market data related to interest rate swaptions from these two markets as of 4 Feb 2020.

The thesis is comprised of four chapters. The first chapter tackles the mathematical preliminaries that are necessary to lay down the theoretical basis and descriptions of money market concepts and interest rate derivatives. The second chapter will cover the classical Black Scholes model and its extension, which is the Black 76 model, as well as the Bachelier model. The third chapter will discuss the SABR model and the effects of its parameters on the volatility smiles/skews. As we are considering the Euribor market swaptions, there is a need to cover a negative rate extension, in particular the shifted SABR model. It is worth mentioning that the modeling will be done using the Hagan's approximation formulas. The last chapter will introduce the market data, calibration concept and the results for two separate markets: Euribor and USD Libor. The fitted models will be evaluated using the "leave one out" estimation approach and several visualisations will be shown and discussed. Finally, in the conclusion key findings from empirical section will be summarized and discussed upon.

1 Mathematical Preliminaries

This chapter is inspired from the literature by Damiano Brigo and Fabio Mercurio [2].

The chapter aims to set up the mathematical basis for the financial market and introduce the relevant notations.

In addition, the basic financial instruments constituting the topic of the interest of this thesis will be defined and explained.

1.1 Money-market Preliminaries

The very beginning of the discussion about the Money-market should start with the definition of the Money-market account.

Definition 1.1 *Money-market (Bank) account:*

We define $M(t)$ to be the value of a money-market account at time $t \geq 0$. We assume $M(0) = 1$ and the money-market account evolves according to the following differential equation:

$$dM(t) = r(t)M(t)dt \tag{1.1}$$

One can solve this differential equation easily and derive the expression for the money-market account.

$$M(t) = \exp \left(\int_0^t r(s)ds \right) \tag{1.2}$$

Formula (1.1) represents the value of the money-market account at time t , where $r(t)$ represents the short-term interest rate(short rate). This can be viewed as the rate at which one can borrow money for an infinitely small period of time.

We know that money has a time based value. The money received today and in a year are not valued the same. There is some relation between the money value of different time moments. This concept is also incorporated in the money-market account idea. We have seen that if we deposit 1 unit of currency at time 0, then at time $t > 0$ we have $M(t)$ units of currency. Therefore, if we desire to have 1 unit of currency at time T , we should invest $A = 1/M(T)$ at time 0. Consequently, the value of A at time t , $0 < t < T$ is $AM(t) = \frac{M(t)}{M(T)}$. This means that the value of 1 unit currency paid at time T can be

seen as $\frac{M(t)}{M(T)}$ at time t . As the rate r has a stochastic nature, the $M(t)$ is also a stochastic process. Let us now define the idea of a discount factor.

Definition 1.2 *Stochastic discount factor:* *The (stochastic) discount factor $D(t, T)$ between two time instants t and T is the amount at time t that is "equivalent" to one unit of currency payable at time T , and is given by*

$$D(t, T) = \frac{M(t)}{M(T)} = \exp\left(-\int_t^T r(s)ds.\right) \quad (1.3)$$

Another significant constituent related to money-market is the zero-coupon bond. Zero-coupon bonds, in case if they are liquid for several maturities, can be the basis to construct the term-structure of interest rates. In addition, Zero-coupon bonds can be used for discounting purposes. In fact, the whole valuation of financial instruments is based on the idea of discounting future cash flows.

Definition 1.3 *Zero-coupon bond:* *A T -maturity zero-coupon bond is a contract that guarantees its holder the payment of one unit of currency at time T . The contract value at time $t \leq T$ is denoted by $P(t, T)$.*

$$P(T, T) = 1 \quad \forall T.$$

We define $R(t, T)$ to be the continuously-compounded interest rate at time t for the period of $[t, T]$.

$$R(t, T) = -\frac{1}{T-t} \ln P(t, T) \quad (1.4)$$

Therefore, the price of a zero-coupon bond can be formulated by following equation:

$$P(t, T) = e^{-R(t, T)(T-t)} \quad (1.5)$$

Having the definition of $R(t, T)$ as (1.4), let us introduce the idea of a short-term rate, which is the limit value expressed by $R(t, T)$: $r(t) = \lim_{T \rightarrow t} R(t, T)$.

Having introduced the ideas of discount factors and zero-coupon bonds, a question arises: What is the relationship between the discount factor $D(t, T)$ and the zero-coupon bond price $P(t, T)$? If rates r are deterministic, then necessarily they are equal. However, if the rates are stochastic, then the discount factor is a random quantity depending on the

future evolution of the interest rates. On the other hand, the zero-coupon bond price has to be known and is a deterministic value. Nevertheless, it can be proven that under a certain probability measure the price of a zero-coupon bond can be the expectation of the discount factor for each pair of time moments (t, T) .

Let us now discuss the term structure of interest rates(also referred as yield curve). The term structure of interest rates describes the relation between interest rates and time to maturities($T - t$). One can expect that the values of interest rates are different for the different time horizons. The term structure of interest rates can be expressed in three ways. Those are the yield curve, discount curve and the forward curve. Among those, the yield curve is the most common way to express it, and it basically represents $R(t, T)$ as a function of t , while the discount curve expresses the relation between zero-coupon bonds and their time to maturity.

In fact, it should be mentioned that there is a market convention to quote yield curve as a combination of money-market and capital-market rates. Money-market rates refer to short-term lending, typically up to one year, while capital market represents the long-term money lending and borrowing typically with the time horizon exceeding one year. Thus, the market standard is to quote the rates of yield curve as simply-compounded rates for the time to maturities up to a year, and the rates for the longer maturities are quoted as compound rates. Let us define the simply-compounded interest rate.

Definition 1.4 *Simply-compounded interest rate.*

Define by $E(t, T)$ the simply-compounded interest rate at time t that applies to the period $[t, T]$ as follows:

$$E(t, T) = \frac{1 - P(t, T)}{(T - t)P(t, T)} \quad (1.6)$$

Thus, the price of the zero-coupon bond can be expressed in terms of simply-compounded interest rate $E(t, T)$

$$P(t, T) = \frac{1}{1 + E(t, T)(T - t)} \quad (1.7)$$

We are going to discuss the forward rates, thus we need to understand the forward-rate-agreement (FRA). All the time variables are being measured in years in this thesis. Forward-rate-agreements are defined by 3 points in time. Those are current time t , the settlement date T_1 and the maturity date T_2 . The following chain of inequalities should

hold $T_2 > T_1 \geq t$. A forward-rate-agreement (FRA) is an over-the-counter contract in which two parties agree to make payments at the maturity date T_2 . The holder of FRA needs to make an interest payment based on a fixed rate r_k and in return receives a floating interest payment based on a spot interest rate, which is the actual rates of a specific floating interest rates observed at a time. The spot rate is set two business days prior to the settlement date T_1 and has maturity T_2 . Thus, it can be expressed by $E(T_1, T_2)$. As a part of a FRA contract the principal amount (L) is mentioned on which the interest rates are applied and the day-count convention is chosen. Let us for now assume that $L = 1$ and there is no day-count convention applied.

We defined already that FRA holder/buyer is the one who pays the fixed leg and the seller pays the floating leg (spot rate fixed at T_1). Now, we can present the cash flow from the holder's perspective at time T_2 as

$$L(E(T_1, T_2) - r_k)(T_2 - T_1) \quad (1.8)$$

For the valuation purposes we want to calculate the value of the contract at time t . That procedure should be done by discounting the value at time T_2 to time t . For this purpose let us focus on the expression

$$(E(T_1, T_2) - r_k)(T_2 - T_1).$$

We can use the definition (1.6) to replace $E(T_1, T_2)$ with $\frac{1-P(T_1, T_2)}{(T_2-T_1)P(T_1, T_2)}$. We will have

$$\left(\frac{1 - P(T_1, T_2)}{(T_2 - T_1)P(T_1, T_2)} - r_k\right)(T_2 - T_1).$$

Opening the brackets will result to the following expression

$$\frac{1}{P(T_1, T_2)} - 1 - r_k(T_2 - T_1).$$

Now, let us find the equivalent value of the above expression at time t . We can do this with in two parts. Firstly, the amount of money $-1 - r_k(T_2 - T_1)$ receivable at time T_2 is equivalent to the following amount of money receivable at time t

$$P(t, T_2)(-1 - r_k(T_2 - T_1))$$

where simply multiplies it with $P(t, T_2)$. Secondly, let us denote the remaining part of the sum by $A = \frac{1}{P(T_1, T_2)}$. If we have A unit of currency at time T_2 , its value at time

T_1 is obtained by multiplying A with a zero-coupon bond price $P(T_1, T_2)$: $P(T_1, T_2)A = P(T_1, T_2)\frac{1}{P(T_1, T_2)} = 1$. Thus having A at time T_2 is equivalent to holding one unit of currency at time T_1 . Additionally, one unit of currency at time T_1 is worth $P(t, T_1)$ units of currency at time t . Therefore, the amount $\frac{1}{P(T_1, T_2)}$ in T_2 is equivalent of $P(t, T_1)$ in t . Combining two parts together we can say that the value of the (1.8) money receivable at time T_2 can be expressed by

$$V_{FRA}(t) = L[P(t, T_1) - P(t, T_2) - P(t, T_2)(T_2 - T_1)r_k] \quad (1.9)$$

The values needed to price the forward contract with the formula (1.9) are the actual forward fixed leg rate and the zero-coupon bond values. which are given with the market yield curve. Currently, having the basic knowledge about the FRA contracts and its value, let us shift to the simply-compounded forward rate.

Definition 1.5 *Simply-compounded forward interest rate.*

*A simply-compounded interest rate is called a simply-compounded **forward** interest rate, if by plugging it into the equation (1.9) the value of the FRA is exactly zero. Thus, it is determined as a root of the equation (1.9) to zero solved for r_k .*

$$F(t, T_1, T_2) = \frac{1}{T_2 - T_1} \left(\frac{P(t, T_1)}{P(t, T_2)} - 1 \right) \quad (1.10)$$

where $t < T_1 < T_2$ and T_1 is the time of expiry, T_2 is the time to maturity.

The next financial instrument to be discussed in the money-market context is the interest rate swap. Similar to a FRA, the swap contract is meant to exchange fixed and floating legs of payments between two parties. Unlike a FRA contract, a swap contract exchanges more than one payment. The counterparty who pays the fixed leg and receives the floating leg is also referred to as a swap payer and the opposite counterparty is called a swap receiver. The payment dates should be predetermined. Let us denote them by T_1, T_2, \dots, T_{mn} . Here m represents the frequency of payments and n denotes the time to maturity of the contract. Thus $m = 1$, if the contract fixes one payment per year. Let us also denote fixed interest rate by r_k (following the option strike notation logic for rates), the principal amount by L and the floating interest rate by $E(T_{i-1}, T_i)$, which is fixed at T_{i-1} and paid

at T_i . According the suggested notation, the floating rates set at the $T_0, T_1, \dots, T_{mn-1}$ succession. Note that there exist swap contracts for which the fixed and floating legs have different time successions, but those are left out from the scope of this thesis. Based on these notations we can formulate the payoff of the swap payer contract as follows:

$$Payoff_{payer} = \sum_{i=1}^{mn} L(T_i - T_{i-1}) (E(T_{i-1}, T_i) - r_k) \quad (1.11)$$

The swap receiver is obvious based on the payer formula.

One needs to discount the cash flows of the swap's payoff in order to get its value at time t .

$$V_{swap}^{payer}(t) = \sum_{i=1}^{mn} P(t, T_i) L(T_i - T_{i-1}) (E(T_{i-1}, T_i) - r_k) \quad (1.12)$$

In addition, just thinking about the swap contract conceptually will hint us that it can be viewed as a combination of several FRA contracts bounded together. Thus, we can rewrite the above equations by using the FRA values. Let us rewrite the formula of a payer swaption's value (1.12) as follows

$$V_{swap}^{payer}(t) = L \sum_{i=1}^{mn} P(t, T_i) L(T_i - T_{i-1}) (F(t, T_{i-1}, T_i) - r_k) \quad (1.13)$$

Now let us substitute $F(t, T_{i-1}, T_i)$ with $\frac{1}{T_i - T_{i-1}} \left(\frac{P(t, T_{i-1})}{P(t, T_i)} - 1 \right)$ as the formula (1.10) suggests. Afterwards, let us note that $\sum_{i=1}^{mn} [P(t, T_i) - P(t, T_{i+1})] = P(t, T_0) - P(t, T_{mn})$, because the values in between the start and end time moments cancel each other out. Applying the previous observations to the (1.13) we can rewrite it as follows

$$V_{swap}^{payer}(t) = LP(t, T_0) - LP(t, T_{mn}) - L \sum_{i=1}^{mn} P(t, T_i) (T_i - T_{i-1}) r_k \quad (1.14)$$

Similarly, the swap receiver's value can be calculated.

Having the analytical formulas for the values of the swap payer contracts makes it straight forward to define the forward swap rate.

Definition 1.6 *Forward swap rate.*

Forward swap rate is the rate r_k which results the value of either payer or receiver swap

to be equal to zero. Therefore, in order to get the forward swap rate (1.14) equation to zero should be solved for r_k , which results to

$$S(t, T_0, T_{mn}) = \frac{P(t, T_0) - P(t, T_{mn})}{\sum_{i=1}^{mn} P(t, T_i)(T_i - T_{i-1})} \quad (1.15)$$

where $S(t, T_0, T_{mn})$ is the forward swap rate at time t , T_0 and T_{mn} are the first reset date and the final payment date respectively.

1.2 Swaptions

Having established some basic understanding about the interest rate market, we can now move on to swaptions, which are the main instrument we will discuss in this thesis. A swaption is an option on a swap. And the option itself is a right to enter into the underlying contract (or to buy/sell the underlying instrument) at a given time and conditions. In our case the underlying is a swap contract. Moreover, we already discussed two sides of a swap, which are swap payer and swap receiver. Thus there are payer swaptions and receiver swaptions. A payer swaption gives the owner the right to enter into a payer swap contract at the option maturity/expiry date T_0 , where the fixed leg is paid and the floating is received. Analogously, the receiver swaption gives the owner the right to enter into a swap at T_0 , where the fixed leg is received and the floating leg is paid.

The swaption maturity is the expiry time of the option. The swaption tenor will denote the length of the underlying swap contract $T_{mn} - T_0$. As the swaption is a right but not an obligation to enter into a swap contract, therefore the payoff of a swaption should be the value of the underlying swap at exercise date of the swaption as long as it is a positive number. Meaning that the swaption will be executed if and only if the underlying swap has a positive value, otherwise the payoff of the swaption will be zero.

$$Payoff_{swaption}^{payer}(t) = \max \left\{ L[P(t, T_0) - P(t, T_{mn})] - L \sum_{i=1}^{mn} P(t, T_i)(T_i - T_{i-1})r_k, 0 \right\} \quad (1.16)$$

The payer or receiver swaption is called at the money (ATM), if at the expiration date T_0 the payoff of the swaption is exactly zero. Similarly, before the expiration date, if the current value of the underlying swap is zero, we still say that the swaption is at the money. Thus, in order for a swaption to be at the money, the rate r_k should be chosen

such that the $V_{swap}^{payer}(t)$ equals zero. Therefore in order to get the rate r_k^{ATM} one should just solve (1.14) equation to zero.

$$r_k^{ATM} = \frac{P(t, T_0) - P(t, T_{mn})}{\sum_{i=1}^{mn} P(t, T_i)(T_i - T_{i-1})} \quad (1.17)$$

We can see that this is exactly the same formula as we derived for the forward swap rate (1.15).

As we might be interested in different expressions of the swaption's payoff or discounted payoff formulas for some valuation purposes, let us go ahead and derive one now. Recall the payer swap's value formula (1.13) based on the concept of it being constituted as a set of FRA contracts. If we encompass it into the scope of an option payoff, we will have the following formula for the payer swaption's payoff at the expiry/maturity time T_0 . Instead of the usual discount factors, here as well, we use the respective bond prices.

$$Payoff_{swaption}^{payer}(T_0) = L \max \left\{ \sum_{i=1}^{mn} P(T_0, T_i)(T_i - T_{i-1})(F(T_0, T_{i-1}, T_i) - r_k), 0 \right\} \quad (1.18)$$

Furthermore, having the formulas (1.10) for the forward interest rate and (1.15) for the forward swap rate, we can make a substitution as follows

$$Payoff_{swaption}^{payer}(T_0) = L \sum_{i=1}^{mn} P(T_0, T_i)(T_i - T_{i-1}) \left[\max \left\{ (S(T_0, T_0, T_{mn}) - r_k), 0 \right\} \right] \quad (1.19)$$

Similar to the payer swaption payoff formulas the receiver swaption payoff formulas are straight forward to derive. As the topic of this thesis is to discuss swaption pricing models and compare them, also derive the volatility surface, the swaption type of interest is going to be the payer swaption. The payer swaption is the instrument of hedging interest rate risk and using this instruments market prices it is possible to have insights into the interest rate risk calculations and hedging. Moreover, for the standard practices such as interest rate model calibration and pricing the European plain-vanilla swaptions are used as a rule, thus we will focus on those during the thesis.

As discussed previously, a swaption is an option on a swap, which gives a right to its owner but not an obligation, therefore it comes with an upfront cost called swaption premium. This premium is a subject of negotiation between the counterparts. The size of the premium depends on whether the swaption is in the money or not and how much it is

in the money at an agreed fixed rate and what the volatility is in the interest rate market. Whether the option is in the money or not depends on the currently agreed fixed rate and the term structure of the interest rates. The maximum loss occurring from holding a swaption can be the premium of the swaption if one chooses not to exercise it.

Let us formulate the In/Out of the money concepts of a swaption. The swaption is said to be in-the-money if r_k^{ATM} is lower than the forward swap rate that applies to the contract and out of the money in the opposite situation.

The value of the swaption depends on the strike rate r_k , the length of the underlying swap, the time to expiration of the option on the swap and the volatility of the interest rates. In order to value swaptions financial engineers either use complex term structure models or the Black-76 model. Black-76 is an accepted industry standard. It is simple and easy to implement. The volatility parameter for Black-76 model is a two-dimensional input parameter, meaning that it depends on the time to maturity of the swap and the expiry date of the option. Later on in this thesis we will also talk about SABR model, which is one of the most complex methods to model interest rate markets, but the next chapter will discuss the industry standard Black-76 model.

2 Black 76 model

This chapter has been written based on the notations and ideas presented in the book "Options, Futures And Other Derivatives" by John Hull, 7th edition, 2008 [3]. Black-76 is the simplest and the most basic model to value swaptions. It is developed by Black in 1976. Due to its simplicity and explainability Black-76 has become an industry standard. Black-76 as well as many other pricing models are based on the stochastic differential equations. However, the Black-76, especially nowadays, is so popular because of the market convention. The current market convention for quoting the interest rate risk is to quote volatilities of the swaptions. Those quoted volatilities are nothing more than just implied volatilities which, if plugged in into Black-76, will return the exact market price of a specific swaption. Basically, the Black-76 is not only a model to price a swaption but also it is there to comfortably report the market quoted swap prices in terms of its volatilities. This phenomenon can take place, because the Black-76 model has only two input variables: volatility and forward rates. As the forward rates are explicitly reported in the market for the current dates, the only variable is left to be the volatility. This proves a one to one mapping between the price of a swaption and the volatility parameter. Summing up, the Black-76 model is absolutely necessary not only for valuation purposes but also for translating the market quoted volatilities into swaption prices. This chapter will introduce the Black-76 model and the scope under which it operates. However, before moving to Black-76 let us initially discuss the basis of Black-76, which is the renowned Black-Scholes option pricing model.

2.1 Black Scholes option pricing model

Black and Scholes developed an option pricing model in 1973 that became the industry standard pricing model for stock options. The model initially was designed to price only non-dividend paying stock options, but later on the model was extended to price also dividend paying stock options. Our thesis is focused on interest rates, thus only the initial non-dividend stock option pricing model will be introduced. That model served as a basis for the Black-76 model, thus this topic will cover it.

Black-Scholes model assumes that the underlying stock price has a stochastic nature and

views it as a stochastic process. Particular assumptions are made regarding the change of the stock price over time. It is assumed that the change of the stock price ΔS over the time period Δt has a normal distribution with $(\mu, \sqrt{\Delta t}\sigma)$ mean and standard deviation. Moreover, it is assumed that this process has Markov properties, particularly it is a memoryless process, for which the past changes cannot affect the future changes and the distributions for each particular non-overlapping periods are independent. Ultimately, the process that is almost surely continuous and its independent increments are normally distributed is called a Brownian motion or sometimes a normal random walk.

Definition 2.1 *A continuous stochastic process W_t ($t \geq 0$) is called a Brownian motion if it satisfies the following properties:*

1. $W_0 = 0$.
2. If $t_1 < t_2 \leq t_3 \leq t_4$ then increments $W_{t_2} - W_{t_1}$ and $W_{t_4} - W_{t_3}$ are independent,
3. $W_t - W_s \sim N(0, t - s)$ for $t \geq s \geq 0$.

The definition above brings up the Brownian motion with a drift rate of zero. In this particular case the expected value of the Brownian motion at any time in the future is exactly the observed value at the current moment. Thus, in order to tackle the specifics of the drift μ , we will need to construct a stochastic differential equation.

$$dX_t = \mu_t(t, X_t)dt + \sigma_t(t, X_t)dW_t \quad (2.1)$$

The SDE consists of a deterministic part $(\mu_t(t, X_t)dt)$ and the stochastic part $(\sigma_t(t, X_t)dW_t)$. The stochastic nature is driven by the W_t . Additionally, μ and σ account for the drift and the volatility respectively. Note, they both are functions of time and the stock price initially.

Now, coming back to the Black-Scholes case specifically, here is what the equation looks in Black-Scholes case

$$dS_t = \mu S_t dt + \sigma S_t dW_t \quad (2.2)$$

where W_t is a Brownian motion. We can go ahead and apply the Ito's lemma and it can be shown that $\ln(S_t)$ follows the process below. Note, the Ito's lemma can be found in book [[3] pp. 269-270]. We will have

$$d\ln(S_t) = \left(\mu - \frac{1}{2}\sigma^2\right)dt + \sigma dW_t \quad (2.3)$$

The process in (2.2) is a Geometric Brownian motion. Thus the stock price follows the Geometric Brownian motion process. However, it is worth mentioning that the logarithmic transformation $\ln(S_t)$ obviously has a constant drift and volatility parameter and is a linear process w.r.t the Brownian motion part. Hereby we have

$$\ln(S_T) \sim N\left(\ln(S_0) + \left(\mu - \frac{1}{2}\sigma^2\right)T, \sigma^2 T\right) \quad (2.4)$$

where S_T is the stock price at some time T in the future and S_0 is the stock price at time zero. As $\ln(S_T)$ has a normal distribution, S_T follows a lognormal distribution by definition. Moreover, the following properties are well known characteristics of lognormal distribution.

$$S_T \sim \text{LogNormal}\left(\ln(S_0) + \left(\mu - \frac{1}{2}\sigma^2\right)T, \sigma^2 T\right) \quad (2.5)$$

$$E(S_T) = S_0 e^{\mu T} \quad (2.6)$$

$$\text{var}(S_T) = S_0^2 e^{2\mu T} (e^{\sigma^2 T} - 1) \quad (2.7)$$

The basic frame of the Black-Scholes model is given. It is also worth mentioning that most of the valuation that is done with the help of this model is based on the risk-neutral pricing theory, which will be tackled in the subsequent subsection. Meanwhile we can move on to the option pricing formula.

The pages 286-287 in the Hull's book bring several assumptions which are necessary to derive the Black-Scholes option pricing formulas. These are:

1. The risk-free interest rate, r^f , is known and is constant through time.
2. The stock price follows the process in (2.5), where μ does not have to necessarily be constant.

3. No dividend paid.
4. There are no transactions costs or taxes in buying or selling the stock or the option.
5. Short-selling is allowed.
6. The market has no arbitrage.
7. Security trading can be done continuously and any fraction of a security can be bought or sold.

Having those assumptions in place, let us consider a European call option. Its payoff at time T is given by $\max(S_T - K, 0)$, where K is the strike price. It has been proven that in case the assumption hold, the fair price of the European call option with the time to maturity T is given with this formula.

$$\begin{aligned}
V_{option} &= S_0 \Phi(d_1) - e^{-r^f T} K \Phi(d_2) \\
d_1 &= \frac{\ln(S_0/K) + (r^f + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \\
d_2 &= d_1 - \sigma\sqrt{T}
\end{aligned} \tag{2.8}$$

Φ is the CDF of standard normal distribution.

2.2 Bachelier's model

Following the same notation and assumptions, let us briefly introduce a very similar model to Black-Scholes, which is called Bachelier's model. The model has been looked up from [6] Iwasawa book. Under Bachelier's model it is assumed that financial quantity follows the SDE:

$$dS_t = \mu dt + \sigma dW_t \tag{2.9}$$

where W_t is a Brownian motion. This is the alternative equation to the Black-Scholes' (2.2). Consequently, this is the core difference between those two modeling approaches. Under Bachelier's model, the European call option's price can be expressed with the following formula

$$\begin{aligned}
V_{option} &= (S_0 - e^{-r^f T} K) \Phi(-d) + \sigma\sqrt{T-t} \phi(d) \\
d &= \frac{K - S_0}{\sigma\sqrt{T-t}}
\end{aligned} \tag{2.10}$$

where Φ and ϕ are the CDF and PDF of standard normal distribution. The proof can be found in [6].

2.3 Risk-neutral valuation

Risk-neutral valuation is a concept used in Black-Scholes model. In risk-neutral world the investors are insensitive towards risk. All the investment possibilities have the same expected value, and that is what the investors rely on while considering investment strategies. The risk-free rate r^f is fixed and all the possible valid investments in the risk-neutral world have the expected return equal to the risk-free rate. Importantly, this only can be true as long as the money-market account is used as a numeraire. On the contrary, the real-world is considered to be risk-averse. The investors require additional risk premium for risky investments. Particularly, in case of the money-market numeraire, the calculation introduced in the risk-neutral world are also valid in the real-world.

The motivation behind using the risk-neutral valuation in the first place comes from the fact that investors in real world are risk-averse, but they all have a specific levels of risk-aversion and the only reasonable way to construct a platform where all the investment strategies can be valued unanimously or at least be agreed upon by the most is the risk-neutral platform. This enables the investors to compare the investment strategies properly. The mathematical basis for this claim is packed within the fundamental theorem of asset pricing, which will be introduced shortly after. Meanwhile, let us define the arbitrage related term.

Let us consider a self-financing portfolio/strategy with a value of V_t at time t (a strategy for which no money is added or withdrawn from the portfolio and all the transactions are financed by selling the existing assets). An arbitrage opportunity is an opportunity for an investor to generate money by starting with a portfolio valued zero, otherwise making money by risking nothing.

1. $V_0 = 0$
2. $V_t \geq 0$
3. $P(V_t > 0) > 0$

As stated previously, we should define the fundamental theorems of asset pricing, but let us firstly go through the ideas of numeraires and martingales. A numeraire is a non-negative asset within the defined set of assets of a particular construct of a market. It serves as a basic asset by which the values of other assets existing in the same market construct are quoted or measured. It is quite common to take the money-market account as a numeraire for pricing purposes. Importantly, a self-financing portfolio remains self-financing even if it is normalized by a numeraire. The next important topic is the concept of a martingale which is a vital part of the arbitrage-free models of financial derivatives. A martingale is a process that is measurable w.r.t. \mathcal{F}_t and the conditional expectation, given all past observations, is equal to the last observation. Following that idea, X_T process will be a martingale w.r.t the \mathcal{F}_t filtration if for each $t \leq T$ follows:

1. $E(|X_T|) < \infty$
2. $E(X_T|\mathcal{F}_t) = X_t$

As we see from the definition, a martingale has no drift.

Assuming that relative price processes are martingales w.r.t some probability measure we can start with the definition of the First Fundamental Theorem of Asset Pricing (FFTAP). The remaining definitions and formulas are inspired by Schumacher, J.M.'s definitions [5].

Definition 2.2 *First Fundamental Theorem of Asset Pricing (FFTAP).*

A market given by a probability measure \mathcal{P} and a set of asset price processes S_1, \dots, S_T is arbitrage-free under the probability measure \mathcal{P} if and only if given any numeraire N , there exists a probability measure \mathcal{Q}_N such that:

1. \mathcal{P} and \mathcal{Q}_N are equivalent probability measures¹.
2. The discounted price processes $\frac{S_1}{N}, \dots, \frac{S_T}{N}$ are martingales under \mathcal{Q}_N .

Returning back to the risk-neutral valuation concept, let us mention that the risk-neutral valuation is done under the risk-neutral probability measure. The latter is an equivalent

¹Two probability measures \mathcal{P} and \mathcal{Q} and are called equivalent if they agree on which events can occur and cannot occur. More specifically, any event A that occurs with positive probability under \mathcal{P} also has to occur with positive probability under \mathcal{Q} . Note that the probability that A occurs may very well differ under \mathcal{P} and \mathcal{Q} .

martingale measure, which is, in fact, any measure that satisfies two conditions stated in 2.2 definition of FFTAP. Moving on, a complete market concept is tackled in the Second Fundamental Theorem of Asset Pricing.

Definition 2.3 *Second Fundamental Theorem of Asset Pricing (SFTAP).*

A market given by a probability measure \mathcal{P} and a set of asset price processes S_1, \dots, S_T and a risk-free bond B is complete under the probability measure \mathcal{P} if and only if there exists a unique risk-neutral measure that is equivalent to \mathcal{P} and has a numeraire B .

Here comes the question of choosing an equivalent martingale measure to be used as a risk-neutral measure. Subsequently, a pricing formula for an attainable cash flow is derived by Schumacher which is called a numeraire dependent pricing formula (NDPF).

$$\pi_t = E_t^{\mathcal{Q}_N} \left[N_t \frac{h(X_T)}{N_T} \middle| \mathcal{F}_t \right] \quad (2.11)$$

where π_t is the value of attainable contingent claim X_t at time t , $h(X_T)$ is the payoff of the contingent claim at time T , N represents any numeraire and $E_t^{\mathcal{Q}_N}$ stands for the expectation under the probability measure which has N asset as numeraire. Additionally, \mathcal{F}_t is the natural filtration or the information available up to time t . Now, naturally, to fit our subject of interest, we should consider the money-market account as a numeraire. Thus, let us see the formula in that case as follows

$$\pi_t = E_t^{\mathcal{Q}_M} \left[M_t \frac{h(X_T)}{M_T} \middle| \mathcal{F}_t \right] = E_t^{\mathcal{Q}_M} \left[e^{-\int_t^T r_s ds} h(X_T) \middle| \mathcal{F}_t \right] \quad (2.12)$$

where $E_t^{\mathcal{Q}_M}$ denotes the expectation at time t under the risk-neutral probability measure which has the money-market account as a numeraire. Thus, it is clear why the expectation represents the discount factor. However, one can use a different numeraire, and in fact, the Black-76 model uses a portfolio of zero-coupon bonds as a numeraire, which we will see in the next subsection.

2.4 Black 76 model for swaptions

As a very well known extension to the classical Black-Scholes option pricing model, the Black-76 model was developed specifically for pricing interest rate derivatives by Fisher

Black in 1976. While these models are quite similar, they still have some objective differences. Firstly, it is worth mentioning that pricing of interest rate derivatives involves more complexity than the pricing of stock options. Secondly, Black-76 model requires the availability of the forward-swap rates, moreover, these and not only these rates are used both for calculating the payoff and for discounting purposes. On the top of this, there is an extra assumption about the nature of the forward-swap rate, specifically an assumption holds of it to be lognormally distributed.

While Black-76 model can be used for pricing interest rate caps, floors, caption, swaptions, let us look only into the last case. Let us consider a payer swaption and try to see the link between the Black-Scholes and Black-76 models, namely by examining the relation of a stock and a swap as underlying asset types. Let us assume that the swap rate is $S(t, T_0, T_{mn})$ as of date t . According to the payer swaption contract the holder of the contract can enter into a swap agreement where the holder will pay the fixed rate and receive the floating rate. Thus, the only scenario under which the swaption will be exercised is when the floating rate is higher than the fixed rate. Subsequently, we can write the payoff of this swaption as follows

$$\frac{L}{m} \max\{S(t, T_0, T_{mn}) - r_k, 0\} \quad (2.13)$$

L is the notional and m indicates the number of payments per year. Now, let us start tackling the pricing problem by firstly introducing the risk-neutral measure. As discussed in the previous subsection, the numeraire does not have to be the money-market account. Moreover, the following expression is used as a numeraire in this model.

$$N_t = L \sum_{i=1}^{mn} P(t, T_i)(T_i - T_{i-1}) \quad (2.14)$$

Let us also denote the expectations at time t under this measure by $E_t^{\mathcal{Q}^N}$. The value of the payer swaption, based on the swap value shown in the previous chapter, will be the following expression

$$payoff_{swaption}^{payer}(T_0) = \max\{L[1 - P(T_0, T_{mn})] - L \sum_{i=1}^{mn} P(T_0, T_i)(T_i - T_{i-1})r_k, 0\} \quad (2.15)$$

Let us make it simple and denote the payer swaption payoff at time t by π_t . Now, recalling the NDPF formula by Schumacher and applying it here will yield the following expressions:

$$\begin{aligned}
& \frac{\pi_t}{L \sum_{i=1}^{mn} P(t, T_i)(T_i - T_{i-1})} = \\
& = E_t^{\mathcal{Q}_N} \left[\frac{\max\{L[1 - P(T_0, T_{mn})] - L \sum_{i=1}^{mn} P(T_0, T_i)(T_i - T_{i-1})r_k, 0\}}{L \sum_{i=1}^{mn} P(T_0, T_i)(T_i - T_{i-1})} \middle| \mathcal{F}_t \right] = \\
& = E_t^{\mathcal{Q}_N} \left[\max \left\{ \frac{1 - P(T_0, T_{mn})}{\sum_{i=1}^{mn} P(T_0, T_i)(T_i - T_{i-1})} - r_k, 0 \right\} \middle| \mathcal{F}_t \right] = \\
& = E_t^{\mathcal{Q}_N} \left[\max \{S(T_0, T_0, T_{mn}) - r_k, 0\} \middle| \mathcal{F}_t \right]
\end{aligned} \tag{2.16}$$

The last expression indicates that the swaption can truly be regarded as a European call option on a swap contract under the N_t (2.14) fixed as a numeraire. Moreover, the forward swap rate can be proven to be a martingale under the specified numeraire. Let us also recall the formula for the forward swap rate, which was used for the derivation of the third expression from the second one in (2.16).

$$S(t, T_0, T_{mn}) = \frac{P(t, T_0) - P(t, T_{mn})}{\sum_{i=1}^{mn} P(t, T_i)(T_i - T_{i-1})} \tag{2.17}$$

Please note that the forward swap rate also can be considered as a tradable asset, namely $\frac{L[P(t, T_0) - P(t, T_{mn})]}{T_{mn} - T_0}$, divided by the numeraire N_t .

We can rewrite the last row of (2.16) for convenience.

$$\pi_t = L \sum_{i=1}^{mn} P(t, T_i)(T_i - T_{i-1}) E_t^{\mathcal{Q}_N} \left[\max \{S(T_0, T_0, T_{mn}) - r_k, 0\} \middle| \mathcal{F}_t \right] \tag{2.18}$$

Now, using the prior knowledge about the Black-Scholes model, we can formulate Black-76 model. Having an assumption of the forward swap rate's distribution being Geometric Brownian motion under \mathcal{Q}_N and the volatility being constant over time, the formula of a payer swaption value according the Black-76 model is stated accordingly.

$$\begin{aligned}
V_{swaption}(t) &= L \sum_{i=1}^{mn} P(t, T_i)(T_i - T_{i-1}) [S(t, T_0, T_{mn}) \Phi(d_1) - r_k \Phi(d_2)] \\
d_1 &= \frac{\ln(S(t, T_0, T_{mn})/r_k) + \frac{1}{2}\sigma^2 T}{\sigma\sqrt{T}} \\
d_2 &= d_1 - \sigma\sqrt{T} \\
T &= T_0 - t
\end{aligned} \tag{2.19}$$

Following the same logic and recalling Bachelier's model (2.10), we can adjust it for the swaption domain as follows:

$$\begin{aligned}
V_{swaption}(t) &= L \sum_{i=1}^{mn} P(t, T_i) (T_i - T_{i-1}) \sigma \sqrt{T-t} (d\Phi(d) + \phi(d)) \\
d &= \frac{S(t, T_0, T_{mn}) - r_k}{\sigma \sqrt{T-t}}
\end{aligned} \tag{2.20}$$

3 SABR model

This chapter is going to be based on the original paper [1] "Managing Smile Risk" by Patrick Hagan and others.

European options are frequently priced and hedged by Black-Scholes model or Black's model. In Black's model there is a direct link between the volatility parameter (let us denote it as σ_B to indicate that we refer to Black's volatility) and the option price. The model itself is quite simple and having the σ_B parameter we can price the option easily. Similarly, we can perform the reverse operation, namely, having the market price of an option, we can uniquely derive the volatility under which the model will result the price equal to the market price, thanks to the monotonic increasing nature of the option price w.r.t. volatility parameter. Subsequently, we will call this volatility an implied volatility. In fact, in many markets it is a standard practice to quote option prices in terms of Black's implied volatility. The construct of Black's formula assumes that σ_B parameter is a constant across all strikes and maturities for each underlying. However, the implied volatility nearly always varies across the strike and maturity dimensions. Thus, when plotting all the implied volatility parameters over all possible strikes we get a graph that is called a volatility smile. This inherent inconsistency sometimes leads to several problems in managing large books of options or pricing some exotic options.

One example can be the pricing of an option with strike K_1 which has a down-and-out knock-out at $K_2 < K_1$. Here a natural question rises. Should we use the implied volatility for K_1 or K_2 or their mean? Another problem can be seen in delta and vega hedging strategies. Thus, the problem can be solved only if we have a single self-consistent model that works for all strikes. An apparent solution to this was introduced with the local volatility models, where the surface between the implied volatility points was approximated based on some logic. The logic was usually taken to be a piecewise constant function. Nevertheless, as mentioned in the paper, this models are giving some counter intuitive and empirically wrong dynamical changes when applied. As a solution to these problems SABR model is introduced by the authors of the paper.

3.1 SABR model

Instead of basing the model on one Brownian motion the authors of SABR model decided to consider the volatility parameter as a separate stochastic process. The decision is justified by the observed chaotic periods in the market, which suggest that the volatility is a stochastic process itself. The Stochastic- $\alpha\beta\rho$ model is introduced as follows

$$\begin{aligned} dF &= \alpha F^\beta dW_1, & F(0) &= f \\ d\alpha &= \nu \alpha dW_2, & \alpha(0) &= \alpha \\ dW_1 dW_2 &= \rho dt \end{aligned} \tag{3.1}$$

It models the forward price F , W_1 and W_2 are brownian motions with correlation ρ . α is a volatility parameter, which in turn is a stochastic process with its own volatility parameter ν , it is called volatility of volatility.

The authors of the paper propose that SABR can be a very good tool to fit the volatility curves observed in the market for any single exercise date with a very high accuracy. However, they also mention that for some underlying instruments SABR model is not the best choice, pointing out that the model is not very useful for the underlying instruments for which the exercise dates can be more than one. These type of instruments include FX rates and most of the stocks. Therefore, the perfect environment where SABR can perform the finest is the interest rate market. And the prominent instruments within the interest rate market are swaptions, caplets and floorlets. The paper introducing SABR shows that by using a singular perturbation techniques, it is possible to derive an approximate solution for the model. Moreover, the upshot of the paper's analysis is that the solution is actually returned as a volatility estimate which then needs to be fed into the Black's formula to get the price of the option. This approach is chosen to adjust to the market convention, which is to quote prices in terms of Black's volatility. Similarly, another formula is derived for quoting the Bachelier volatility parameter, which in turn should be used in Bachelier's model to price an option. The latter will be mentioned later on. Currently, having the Black's formulas (2.8) the following expression can be used to approximate the Black's volatility parameter $\sigma_B(K, f)$, where K is the strike price and f

is the forward price of the underlying.

$$\sigma_B(K, f) = \frac{\alpha}{(fK)^{(1-\beta)/2} \left\{ 1 + \frac{(1-\beta)^2}{24} \ln^2 f/K + \frac{(1-\beta)^4}{1920} \ln^4 f/K + \dots \right\}} \cdot \left(\frac{z}{x(z)} \right) \cdot \left\{ 1 + \left[\frac{(1-\beta)^2}{24} \frac{\alpha^2}{(fK)^{1-\beta}} + \frac{1}{4} \frac{\rho\beta\nu\alpha}{(fK)^{(1-\beta)/2}} + \frac{2-3\rho^2}{24} \nu^2 \right] T + \dots \right\} \quad (3.2)$$

where

$$z = \frac{\nu}{\alpha} (fK)^{(1-\beta)/2} \ln(f/K) \quad (3.3)$$

and

$$x(z) = \ln \left\{ \frac{\sqrt{1 - 2\rho z + z^2} + z - \rho}{1 - \rho} \right\} \quad (3.4)$$

If we consider a special case, when the option is at the money (a.k.a. $K = f$), we can reduce the formula down to the following

$$\sigma_{ATM}^B = \sigma_B(f, f) = \frac{\alpha}{f^{(1-\beta)}} \left\{ 1 + \left[\frac{(1-\beta)^2}{24} \frac{\alpha^2}{f^{2-2\beta}} + \frac{1}{4} \frac{\rho\beta\alpha\nu}{f^{(1-\beta)}} + \frac{2-3\rho^2}{24} \nu^2 \right] T + \dots \right\} \quad (3.5)$$

The omitted terms " $+\dots$ " are much smaller. Indeed, even though more accurate expressions were derived in the original paper by continuing the perturbation expansion to higher order, the (3.2) and (3.5) are the formulas being used to value and hedge vanilla swaptions, caps and floors. The higher order results have not been implemented, believing that the increased precision of the higher order results is superfluous. (Hagan *et al.*, 2002)

There are two special cases to be noted: $\beta = 1$ represents the stochastic log-normal model and $\beta = 0$ normal model. The implied volatility formulas for these special cases will be introduced later.

Similar to the Black's implied volatility formula, Patrick Hagan has derived an implied volatility formula for Bachelier's model. It is also referred to as a Normal volatility formula and is presented as follows

$$\sigma_N(K, f) = \alpha(fK)^{\beta/2} \frac{1 + \frac{1}{24} \ln^2 f/K + \frac{1}{1920} \ln^4 f/K + \dots}{1 + \frac{(1-\beta)^2}{24} \ln^2 f/K + \frac{(1-\beta)^4}{1920} \ln^4 f/K + \dots} \cdot \left(\frac{z}{x(z)} \right) \cdot \left\{ 1 + \left[\frac{-\beta(2-\beta)\alpha^2}{24(fK)^{1-\beta}} + \frac{\rho\alpha\nu\beta}{4(fK)^{(1-\beta)/2}} + \frac{2-3\rho^2}{24} \nu^2 \right] T + \dots \right\} \quad (3.6)$$

where z and $x(z)$ are the expressions from (3.3) and (3.4). Here the at the money case can also be shown as a simplified expression like

$$\sigma_{ATM}^N = \sigma_N(f, f) = \alpha f^\beta \left\{ 1 + \left[\frac{-\beta(2-\beta)\alpha^2}{24(f)^{2-2\beta}} + \frac{\rho\beta\alpha\nu}{4f^{(1-\beta)}} + \frac{2-3\rho^2}{24}\nu^2 \right] T + \dots \right\}. \quad (3.7)$$

3.2 Breaking down SABR parameters

As the underlying's forward price varies throughout the trading day, the curve that ATM volatility (3.5) $\sigma_B(f, f)$ traces is known as the backbone, while the smile and skew refer to the implied volatility $\sigma_B(K, f)$ as a function of strike K for a fixed forward price f .

The backbone is affected by the parameter β . While the backbone curve takes a downward sloping shape when $\beta = 0$, it starts to become less and less downward when β increases and becomes completely horizontal at $\beta = 1$.

Let us now see how each part of the expression affects the shape of the volatility smile. For that reason, let us introduce a slightly simplified version of the volatility expression (3.2).

$$\begin{aligned} \sigma_B(K, f) = \frac{\alpha}{f^{1-\beta}} & \left\{ 1 - \frac{1}{2}(1 - \beta - \rho\lambda) \ln K/f \right. \\ & \left. + \frac{1}{12}[(1 - \beta)^2 + (2 - 3\rho^2)\lambda^2] \ln^2 K/f + \dots \right\} \end{aligned} \quad (3.8)$$

where

$$\lambda = \frac{\nu}{\alpha} f^{1-\beta} \quad (3.9)$$

provided that the strike K is not too far from the forward price f .

While this version will not be very accurate, it is still enough to understand the dependency of the implied volatility from the parameters more easily.

As discussed previously, the volatility smile gives a snapshot of the market volatility (equivalently the prices) for different strikes for a given forward price. Now, looking into the simplified volatility expression (3.8) let us break it down to the core. The first factor $\frac{\alpha}{f^{1-\beta}}$ is roughly the implied volatility for ATM options. Thus, the backbone mainly relies on this part, hence, the relation mentioned w.r.t. β is justified. The second term $\frac{1}{2}(1 - \beta - \rho\lambda) \ln K/f$ stands for the skew, the slope of the implied volatility curve w.r.t. the strike price. This effect, in particular, can be split into two effects: $-\frac{1}{2}(1 - \beta) \ln K/f$

known as *beta* skew (it is downward sloping since $0 \leq \beta \leq 1$) and $\frac{1}{2}\rho\lambda\ln K/f$ known as the *vanna* skew, which is caused by the correlation parameter ρ . Empirically, that correlation has mostly been negative, meaning the higher the forward price the lower the implied volatility. The latter results the vanna skew being downward sloping. Finally, the last term, which in turn breaks down to two effects. The first part $\frac{1}{12}(1 - \beta)^2 \ln^2 K/f$ is the smile (quadratic term), but it is dominated by the downward sloping beta skew and at reasonable strikes it just affects that skew to some extent. The second part $\frac{1}{12}(2 - 3\rho^2)\lambda^2 \ln^2 K/f$ is the smile induced by the *volga* (vol-gamma) effect. This smile is caused by the fact that extremely large movements of the forward price happen more often when the volatility α increases and less often otherwise. Thus, the strikes too far from the current forward price represent the high volatility environments.

Let us discuss the parameters' effects separately. For that we will again consider a volatility smile for a fixed forward. The parameter β controls the skew. The same does the parameter ρ and it is quite difficult to differentiate the effects of β and ρ on a static smile. Thus, it is a common practice to usually choose β in advance, based on either the historical observations of the backbone or select based on preferences or a priori considerations. The latter is usually a choice between $\beta = 0$ (Bachelier model, normal distribution), $\beta = 0.5$ (stochastic CIR) or $\beta = 1$ (Black's model, log-normal distribution). $\beta = 0$ is usually preferred by the practitioners working with the Euribor products where the interest rates can go below zero. $\beta = 0.5$ is usually used by the US interest rate desks and $\beta = 1$ is supported by the majority of derivative traders with a very basic and standard underlyings, as it is considered to be more natural model to some people.

Subsequently, we are left with the need to calibrate ρ , α , ν . Luckily, these three parameters are responsible for very specific augmentations on the smile. ρ shapes the curve's skew (see figure 1), α mainly controls the overall height of the curve (see figure 2) and ν change controls how much smile the curve exhibits (see figure 3). As the three parameters have a strictly separated roles in the model, it should be easier to maintain stable calibration results despite the market noise, which occur mostly on the far away strikes.

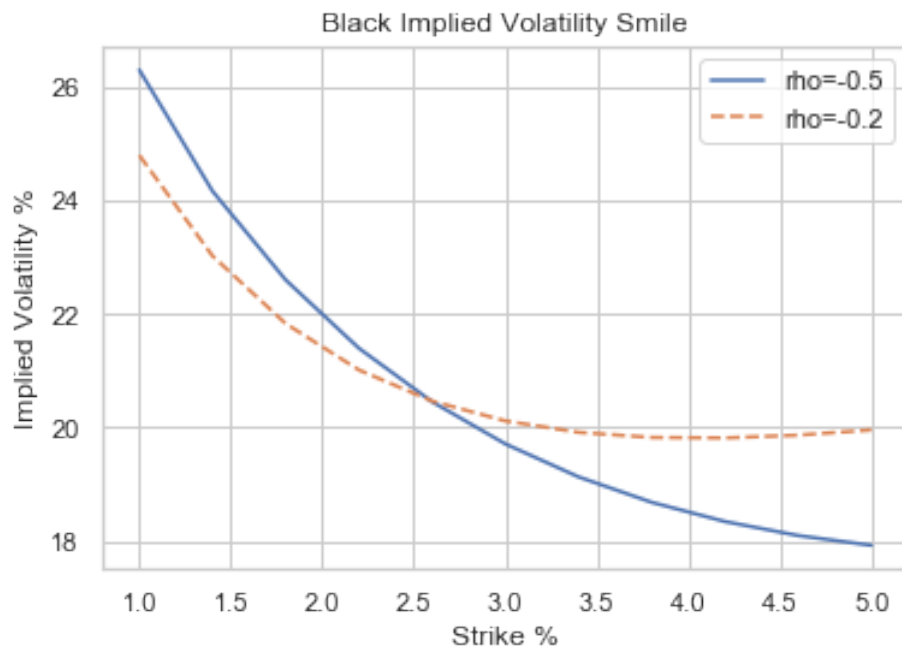


Figure 1: An example of the curve change when ρ is changed, while other parameters are constant.

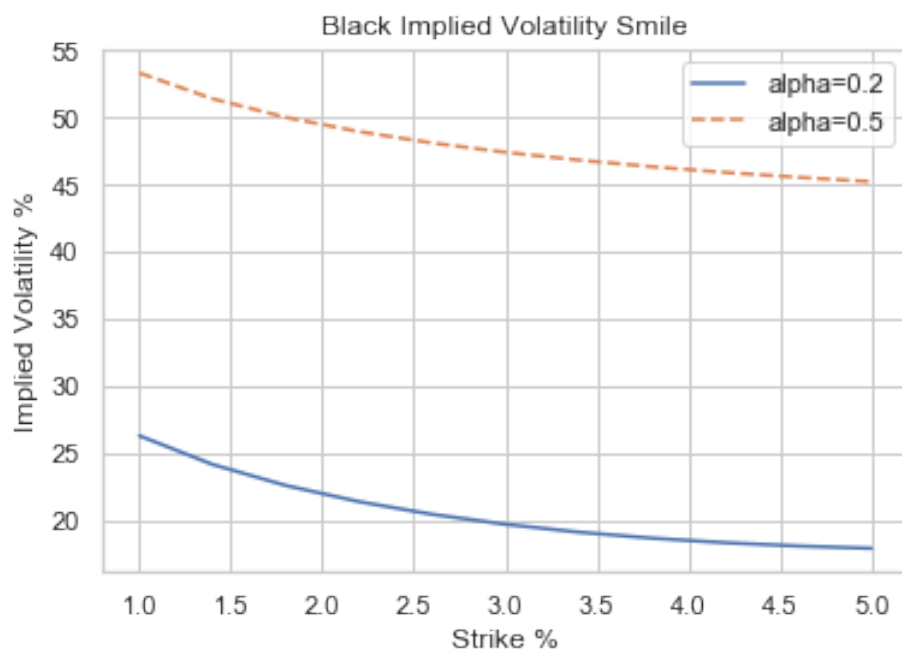


Figure 2: An example of the curve change when α is changed, while other parameters are constant.

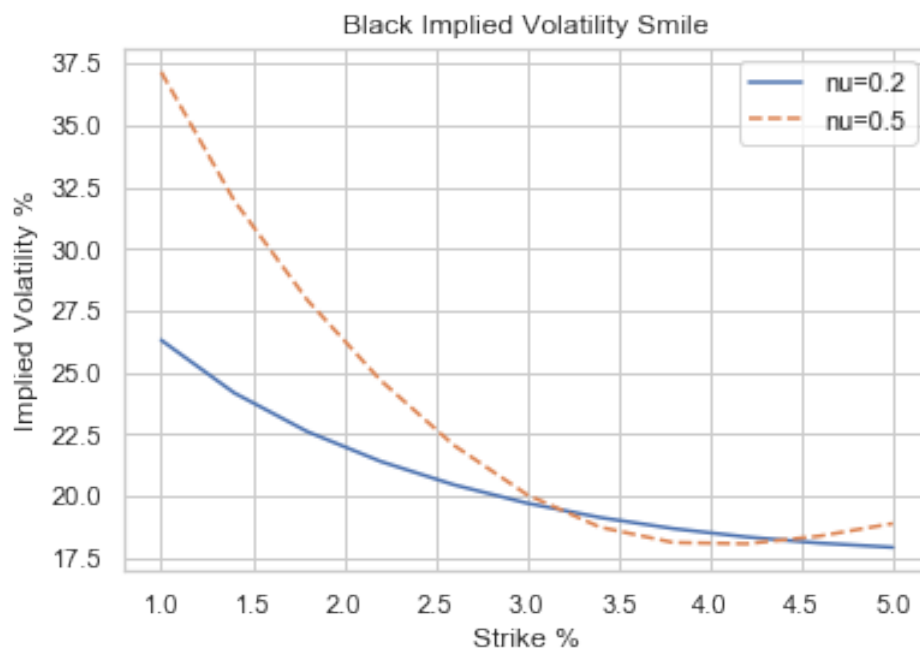


Figure 3: An example of the curve change when ν is changed, while other parameters are constant.

3.3 Shifted SABR model

The Hagan's approximation formulas for Black and Normal implied volatility estimation (3.2) and (3.6) are quite useful and very well accepted among practitioners. However, they have some drawbacks as well. In the current situation of the negative Euribor rates, Hagan's formulas become invalid, because of the logarithms in them. Thus, several approaches have been developed to overcome this issue. Some of them are quite straightforward and some are complex. Among the more complex approaches is the Free Boundary SABR model proposed by Alexandre Antonov in 2015. We will not cover that approach in this thesis. However, we will cover a simpler and more straight-forward approach called Shifted SABR model. It has some benefits and drawbacks, which we will discuss in this chapter.

Shifted SABR is the first and simplest extension of SABR model towards the negative interest rates. It inherits all the advantages and intuitive parameters of the SABR model. Similar to the original SABR (3.1), under the shifted SABR model the forward rate has

the following dynamics:

$$\begin{aligned}
dF &= \alpha(F + s)^\beta dW_1, \quad F(0) = f \\
d\alpha &= \nu\alpha dW_2, \quad \alpha(0) = \alpha \\
dW_1 dW_2 &= \rho dt
\end{aligned} \tag{3.10}$$

with one difference: s is a positive deterministic shift.

The idea of the shift is to change the lower boundary from 0 to $-s$. This in turn, will allow the underlying F go negative down till the boundary. The shift parameter s can be chosen either based on the preference or by some calibration process. In this thesis, we will choose $s = 3\%$ in the next chapter for the Euro market.

This choice is justified because of the current situation and expectations of the Euribor curve. The shift is set such that we do not expect Euribor to go below the shift, but on the other hand, the shift itself is not very large.

Nevertheless, in general the shift should be chosen such that it complies with the following:

- $F + s > 0, \forall F$
- $K + s > 0, \forall K$

Going further, we define the shifted forward rate as : $X = F + s$.

Let's take a payoff of a call option with maturity T .

$$\begin{aligned}
\max(F_T - K, 0) &= \max((F_T + s) - (K + s), 0) \\
&= \max(X_T - (K + s), 0)
\end{aligned} \tag{3.11}$$

We can see that the payoff of a call option on F with the strike K is equal to the payoff of call option on X with the strike $K + s$.

Moreover, let us note that $dF = d(F + s)$ for deterministic shift s . Thus, we can rewrite the first equation from (3.10) in terms of X as follows:

$$dX = \alpha X^\beta dW_1$$

resulting to the conclusion, that X follows SABR model dynamics with the same parameters as F :

$$\begin{aligned}
dX &= \alpha X^\beta dW_1, & X(0) &= f \\
d\alpha &= \nu \alpha dW_2, & \alpha(0) &= \alpha \\
dW_1 dW_2 &= \rho dt
\end{aligned} \tag{3.12}$$

As a conclusion, we can have that the call option payoff of a shifted SABR model with the forward f and strike K is equal to the corresponding SABR model call option payoff with forward $f + s$ and strike $K + s$ at the time of maturity. Let us also note that we do not have different approaches or assumptions about the discount process, which means that if the payoffs are equal, then we should apply the exact same principles of discounting on both cases. Therefore, we can also say that the values of these particular options will also be equal.

Now, assuming the forward swap rate follows shifted SABR dynamics, let us derive Hagan's approximation formulas. Let $BS(f, T, K, \sigma)$ denote the Black-Scholes call option price with forward f , time to maturity T , strike K , volatility σ . The price of a call option of F will be given as:

$$BS(f + s, T, K + s, \sigma_B(K + s, f + s))$$

where the volatility is given by Hagan's Black volatility approximation formula as $\sigma_B(K + s, f + s)$, note that the arguments are shifted.

Summing up, we can say that shifted SABR model was quite straight-forward and easy to understand, if we know the SABR model. As discussed previously it has a drawback. We have to fix the shift, which results to some problems. In particular, a change in the shift may be required if the rate goes below the shift. This can result to a jump in calibration parameters, which in turn might affect the Greeks.

4 Results

This section will discuss the SABR model's calibration on the market data. The chosen markets are Euribor and USD Libor. The valuation date is taken to be 4 Feb 2020. Along with the calibration, some market data engineering is also done, which will be discussed under the Euribor subsection.

Let us first discuss the market data.

4.1 Market Data: Volatility Cube

As discussed in the first chapter, in a specific market the swaption is a contract that can be identified by three attributes: time to maturity/expiry time, tenor and the strike. Volatility cube is a set of volatility quotes for each combination of these three attributes. If we denote the expiry, tenor and strike as T_m , T_n and K , we can denote the respective volatility quote as $\sigma_{mn}(K)$. Volatility cube is the data that represents the volatility quotes for several T_m , T_n and K values, where m and n represent the indices of the two dimensions of the volatility cube: expiry and tenor. This directly means that the expiry and tenor are treated as a predefined set of limited values. K , in fact, also is limited in the volatility cube, but it is not limited by SABR model. If we fix the strike to some K , we will get a volatility surface on that level. The fixed strike can be also fixed conceptually, e.g. the fixed strike can be the forward rate for that specific expiry, tenor combination. In that case the volatility surface is called At the money volatility surface. Similarly the fix can be relative to the forward rate, e.g. ATM+50 bps volatility plane or surface. It is worth to note, that the notation in the interest rate domain is different from the stock options' domain, where the volatility surface is the surface on maturity and strike dimension. Alternatively, if we fix the expiry and tenor, we get the well known volatility smile. Ironically, the volatility smiles in the interest rate market usually look like skews and even sometimes they are referred to as a volatility skew.

Here is an example of a volatility cube expressed in the table [1](#).

Other than the volatility cube, we need the market rates. Market rates can be expressed in different ways. The data used in this thesis represents the discount factors of Euribor 6M curve and USD Libor 3M curve as of 4 Feb 2020.

Expiry	Tenor	Strike	Black Vola	Expiry	Tenor	Strike	Black Vola
1M	1Y	0	27.24	1M	7Y	0	49.24
1M	1Y	-50	17.85	1M	7Y	-50	19.13
1M	1Y	-25	8.04	1M	7Y	-25	7.3
1M	1Y	25	-5.85	1M	7Y	25	-3
1M	1Y	50	-8.06	1M	7Y	50	-3
1M	2Y	0	43.15	1M	10Y	0	46.43
1M	2Y	-50	12.67	1M	10Y	-50	16.48
1M	2Y	-25	4.47	1M	10Y	-25	6.28
1M	2Y	25	-1.48	1M	10Y	25	-2.54
1M	2Y	50	-1.16	1M	10Y	50	-2.44
1M	3Y	0	49.53	1M	20Y	0	41.12
1M	3Y	-50	16.5	1M	20Y	-50	13.63
1M	3Y	-25	5.84	1M	20Y	-25	5.16
1M	3Y	25	-1.82	1M	20Y	25	-1.79
1M	3Y	50	-1.24	1M	20Y	50	-1.22
1M	4Y	0	52.02	1M	30Y	0	40.08
1M	4Y	-50	18.95	1M	30Y	-50	12.56
1M	4Y	-25	7.01	1M	30Y	-25	4.78
1M	4Y	25	-2.71	1M	30Y	25	-1.78
1M	4Y	50	-2.62	1M	30Y	50	-1.43
1M	5Y	0	50.67	3M	1Y	0	32.24
1M	5Y	-50	20.81	3M	1Y	-50	19.08
1M	5Y	-25	7.94	3M	1Y	-25	8.51
1M	5Y	25	-3.28	3M	1Y	25	-6.48
1M	5Y	50	-3.32	3M	1Y	50	-10.02

Table 1: Volatility cube for USD libor, volatility is in percentage, additive relative to ATM volatility.

The discount factors are enough to construct the spot rate curve and forward swap rate curve. Additionally, it is an accepted market convention to use Overnight Interest Swap

curves for the discounting of future cash flows. Thus, the data also includes EUR OIS and USD OIS curves as of 4 Feb 2020. Before starting any valuation process, it is important to fix the day-count convention. In this thesis, $ACT/365$ day-count convention is chosen, which means that the time interval between two dates will be calculated by counting the actual days in between those two dates, where we include the first date and exclude the last one, and dividing the result on 365.

Let us now discuss separately EUR and USD calibration cases.

4.2 Euribor volatility cube calibration with SABR

Let us note that similar to Black volatility cube, there can be Bachelier or normal volatility cube quotes in the market. As a matter of fact, the volatility for Euribor is either quoted by normal implied volatility cube or by shifted Black volatility cube. The latter is harder to find in practice. The data for this thesis includes normal volatility cube for Euribor as of 4 Feb 2020. Additionally, it turns out that in order to utilize Hagan's approximation formulas, the rates/strikes cannot go negative. This makes the application for Euribor harder than for USD Libor. As discussed in the chapter covering SABR model extensions on negative rates, the easiest way would be to use shifted SABR model. However, for the calibration of shifted SABR, the cube should express the implied volatilities for shifted Black model.

In order to solve this issue a market data engineering type of approach is undertaken. The goal of that intermediate step is to retrieve the shifted Black volatility cube out of a normal volatility cube for Euribor swaptions. The process can be described as follows:

- calculate Swaption prices for each quoted normal volatility using Bachelier model.
- set up a shifted Black 76 model with 3% shift.
- Imply the volatility for each swaption price using the shifted Black 76 model and a root finding minimization technique.
- Get the volatility cube for 3% shifted Black volatilities for Euribor swaptions.

The python code implementation of this process can be found in the appendix.

This approach is a quick workaround in cases when the market data is not very easy to

get. And in reality, the market data on specific financial products are quite expensive and rare, so this approach can be viewed as a possibility to use financial engineering techniques in order to overcome the data shortage and be more robust in the highly limited environment.

It should also be noted that in order to skip some numerical precision errors, some of the shifted volatility points are omitted from the analysis due to the fact that the swaption prices were either zero or too close to zero ($< 10^{-9}$). The latter is done because of a concern about the numerical operations' precision in the programming environment. The code in appendix bears that procedure.

Next step will be to preform the calibration of the SABR model for each (expiry, tenor) combination and fitting a smile on the strike dimension. It is worth noting that for each T_m, T_n we fit a separate set of parameters α, ν, ρ . In this thesis it was decided to fix the beta as an a priori parameter. Specifically for the Euro market, β is taken to be 1.

Thus, the calibration procedure includes a function that is minimized for each separate combination of expiry and tenor across all strikes. The function to minimize can be expressed as follows:

$$L(\beta, \alpha, \nu, \rho, T_n, \bar{K}, S(t, T_m, T_n)) = \sum_{k \in \bar{K}} (\hat{\sigma}_B(\beta, \alpha, \nu, \rho, T_n, S(t, T_m, T_n), k) - \sigma_B)^2 \quad (4.1)$$

where \bar{K} is the set of available strikes for the expiry T_m and tenor T_n in the volatility cube, $S(t, T_m, T_n)$ is the forward swap rate. is calculated with the Black's volatility approximation formula by Hagan (3.2). Note, we use forward swap rate S as f and the tenor T_n as T in the formula (3.2).

And the calibration process is the multiparameter optimization of the Loss function L :

$$L(\beta, \alpha, \nu, \rho, T_n, \bar{K}, S(t, T_m, T_n)) \xrightarrow{\alpha, \rho, \nu} \min \quad (4.2)$$

Having all the data, we can perform the calibration and report the results in the following table 2. Each row in the table represents a separate volatility smile/skew for a specific combination of expiry and tenor.

In order to see the results better, we can visualize the fitted smiles for each expiry and tenor combination and the actual points of the volatility in the figure 4.

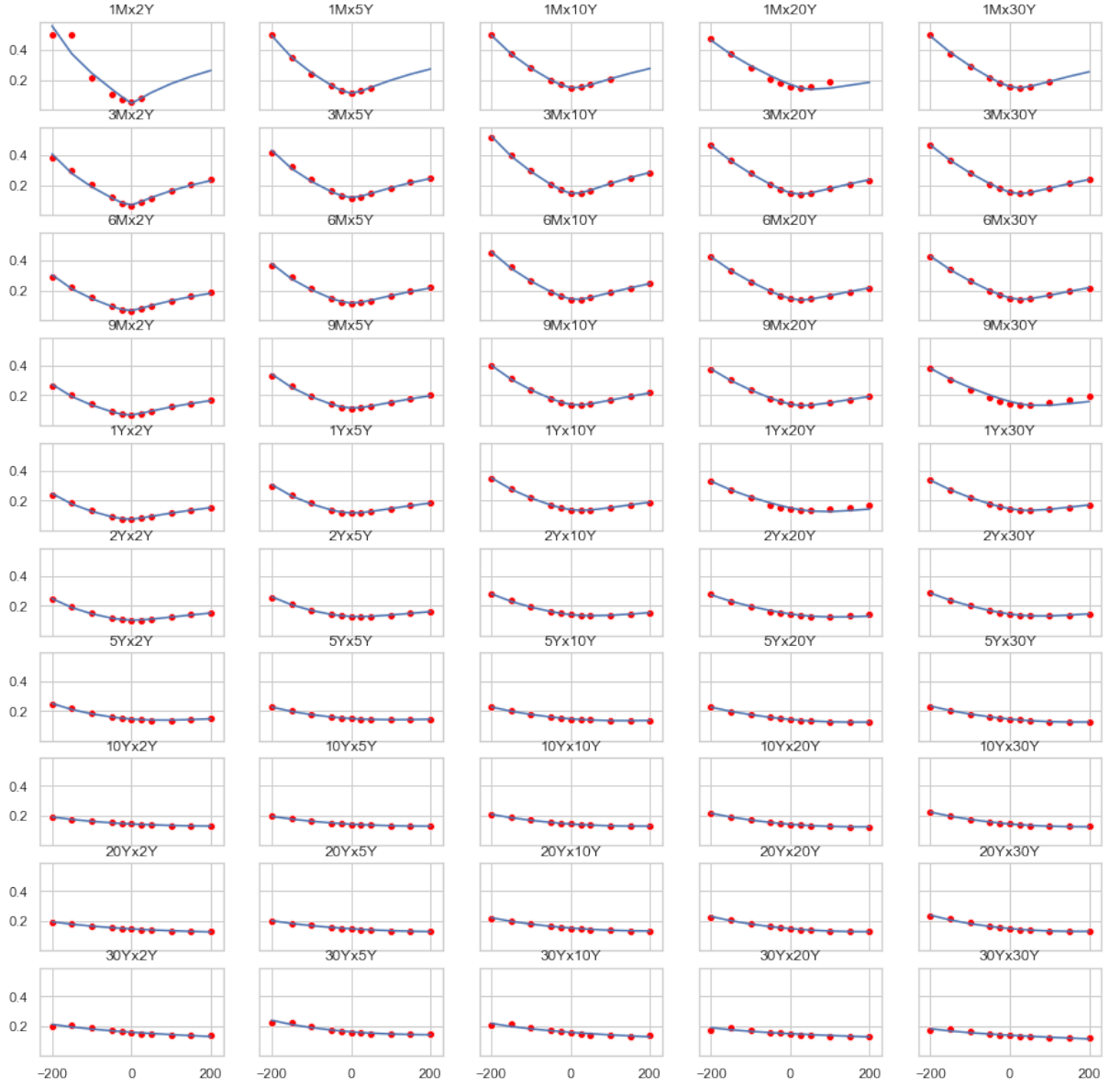


Figure 4: EUR smiles fitted by 3% shifted SABR model. The red dots represent the actual volatility points and the line represents the fitted smile.

Expiry	Tenor	α	ν	ρ	Loss	Expiry	Tenor	α	ν	ρ	Loss
1M	2Y	0.034	1.396	-0.105	0.022	2Y	2Y	0.101	0.438	-0.054	0.000
1M	5Y	0.082	0.932	-0.026	0.000	2Y	5Y	0.122	0.411	-0.157	0.000
1M	10Y	0.099	0.826	-0.173	0.000	2Y	10Y	0.131	0.445	-0.376	0.000
1M	20Y	1.433	8.596	-0.669	0.003	2Y	20Y	0.768	2.347	-0.585	0.000
1M	30Y	0.084	0.728	-0.378	0.000	2Y	30Y	0.129	0.459	-0.488	0.000
3M	2Y	0.059	0.993	0.145	0.001	5Y	2Y	0.145	0.337	-0.357	0.000
3M	5Y	0.093	0.804	-0.032	0.001	5Y	5Y	0.148	0.280	-0.370	0.000
3M	10Y	0.094	0.877	-0.204	0.000	5Y	10Y	0.147	0.304	-0.456	0.000
3M	20Y	0.092	0.772	-0.396	0.000	5Y	20Y	0.622	1.315	-0.561	0.000
3M	30Y	0.087	0.707	-0.399	0.000	5Y	30Y	0.159	0.336	-0.572	0.000
6M	2Y	0.060	0.731	0.172	0.001	10Y	2Y	0.148	0.191	-0.500	0.000
6M	5Y	0.096	0.703	-0.058	0.000	10Y	5Y	0.147	0.226	-0.477	0.000
6M	10Y	0.100	0.784	-0.249	0.000	10Y	10Y	0.149	0.274	-0.493	0.000
6M	20Y	0.095	0.721	-0.400	0.000	10Y	20Y	0.151	0.295	-0.533	0.000
6M	30Y	0.091	0.667	-0.411	0.000	10Y	30Y	0.160	0.305	-0.552	0.000
9M	2Y	0.066	0.629	0.142	0.000	20Y	2Y	0.150	0.165	-0.650	0.000
9M	5Y	0.101	0.627	-0.076	0.000	20Y	5Y	0.152	0.200	-0.590	0.000
9M	10Y	0.107	0.695	-0.262	0.000	20Y	10Y	0.159	0.252	-0.566	0.000
9M	20Y	0.102	0.654	-0.413	0.000	20Y	20Y	0.161	0.284	-0.577	0.000
9M	30Y	0.975	4.656	-0.640	0.003	20Y	30Y	0.164	0.304	-0.554	0.000
1Y	2Y	0.074	0.526	0.109	0.000	30Y	2Y	0.163	0.126	-0.908	0.000
1Y	5Y	0.106	0.547	-0.070	0.000	30Y	5Y	0.166	0.243	-0.544	0.000
1Y	10Y	0.116	0.604	-0.304	0.000	30Y	10Y	0.170	0.155	-0.838	0.001
1Y	20Y	0.849	3.421	-0.622	0.002	30Y	20Y	0.164	0.087	-0.999	0.001
1Y	30Y	0.111	0.545	-0.446	0.000	30Y	30Y	0.165	0.107	-0.999	0.001

Table 2: Euribor calibration results of 3% shifted volatility cube.

From the first glance it looks fine. And the losses are quite well minimized. However, our aim of calibrating the SABR on a volatility cube has a practical reason of filling in the strikes that are not actually present in the market data. Thus, we are interested in how

well the calibrated model can reflect the volatility point in case it was not included in the calibration set.

Aiming to show how the calibrated model behaves on the unseen points, we will perform a "leave one out" approach for each strike point available for each (expiry, tenor) combination and compare the left out strike's actual volatility to the fitted volatility which was calibrated without that point.

The procedure is available in the appendix and the results can be seen with the barplot [5](#).

We can see that the leave one out approach gives in general a quite well fit. Exceptions are the very short expiry and tenor smiles and the strike points that are at the edge, which requires the SABR model to extrapolate, as there is no other points further than that.

Another interesting approach will be to take a surface and see how the fitted surface looks like w.r.t. the actual volatility cube points. The most important surface within the volatility cube is considered to be the at the money surface. This shows all the points where the strike is equal to the respective forward swap rate. Thus, let us look into the fitted ATM surface plot. [6](#).

Note that the volatility surface is derived by performing a linear interpolation on fitted points. The linear interpolation is done to show the surface, otherwise the plot would be a scatter plot and not convenient to observe. Similarly, it is interesting to see the surface on actual points. It is plotted as figure [7](#)



Figure 5: Leave one out approach performed for each strike of each smile on EUR 3% shifted Black volatility cube. The bars on the left are actual values, the bars on the right are fitted values.

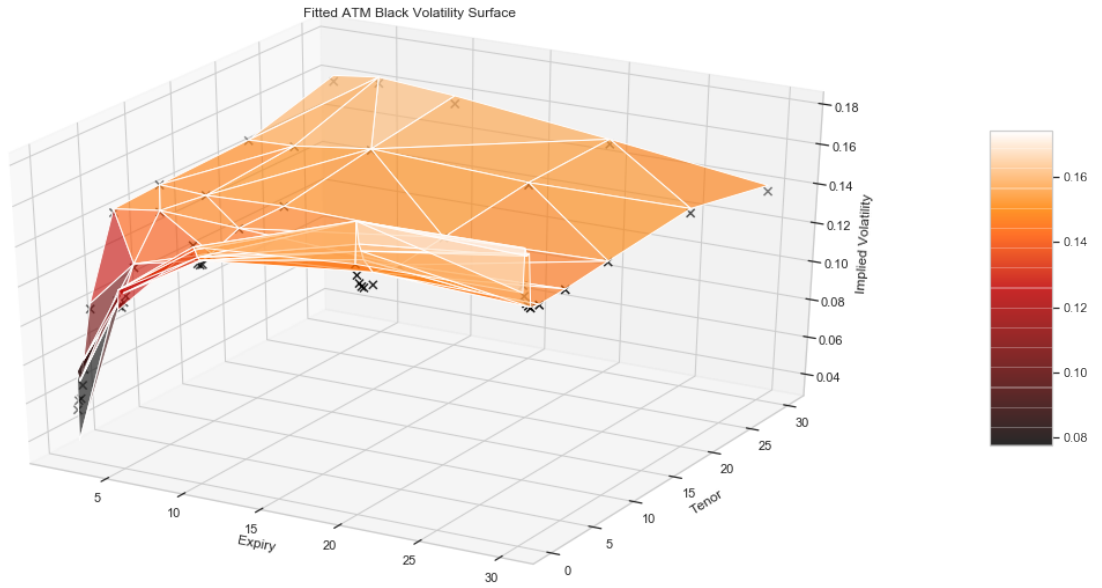


Figure 6: Eur Black fitted volatility ATM surface. The points represent the actual volatility quotes, the surface is fitted.

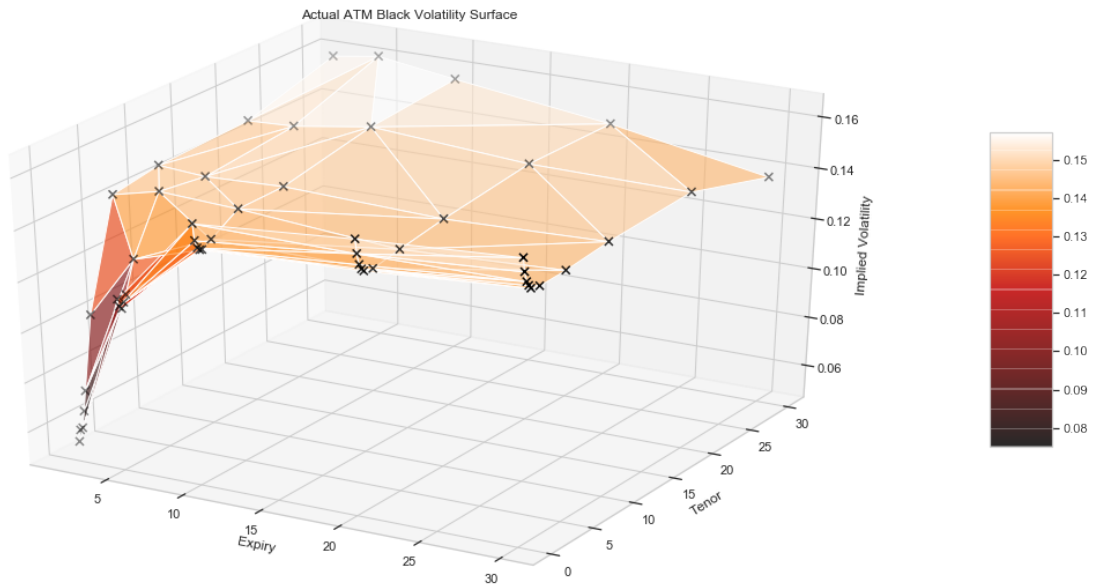


Figure 7: Eur Black actual volatility ATM surface. The points represent the actual volatility quotes, the surface is fitted. Linear interpolation is performed

4.3 USD Libor volatility cube calibration with SABR

In the USD Libor case, the data to be used is Libor 3M curve for forward swap curve derivation, Libor OIS curve for discounting purposes and a Black volatility cube.

In case of Libor, the rates are positive, thus it reduces the amount of work and complexity to get the right data or to use shifted models. Here we will use the straight forward Hagan's approximation formulas for implied Black volatility (3.2). The loss function is the same as in Euribor case (4.1) and we minimize with the same principle (4.2). As the data for Libor appears to be more noisy and volatile in our particular case, a judgement call is made in terms of taking the a priori β parameter. We know that the $\beta = 0$ or $\beta = 1$ are more limited and special cases, while $\beta \in (0, 1)$ is a more general case. Thus, in USD Libor case, the β parameter is taken to be 0.5, which is the middle point of the $(0, 1)$ interval and a more or less standard value for the parameter other than 0 and 1. The code for it is available in the appendix. Having all the data, we can perform the calibration and report the results in the following tables 3 for part 1 and 4 for part 2. The results are broken into two tables because of the size of the Libor data, it happens to be larger compared to the Euribor data.

The results can also be checked by plotting the smiles and the actual points of volatility: figure 8. some expiry, tenor combinations have empty graphs because the market data is not provided for them. However, the aim of plotting them even though the smile does not exist is to keep the tabular form of the possible expiry and tenor time crosses.

Subsequently, in order to test the out of the sample performance of the calibration, similar to the Euribor case, we perform a leave one out approach and plot the barplots 9.

Here as well, we see some odd market data which is a bit hard to fit. Also we observe that the points on the edges of the strike dimension are harder to match than the ones in the middle. The possible reasons of poor fitting capacity for some rare cases are the market noise and the liquidity. Market noise can cause some quotes to have odd values, and the lack of liquidity for some particular positions in the Libor market can cause the volatility quotes to be non reliable. There are some techniques that include adding weights to the quotes based on the bid-ask spread size and some other liquidity indicators. These approaches tend to smoothen the fits and not be affected by outliers. Unfortunately, the data used in this thesis does not cover these liquidity indicators, therefore the above



Figure 8: USD smiles fitted by SABR model. The red dots represent the actual volatility points and the line represents the fitted smile.

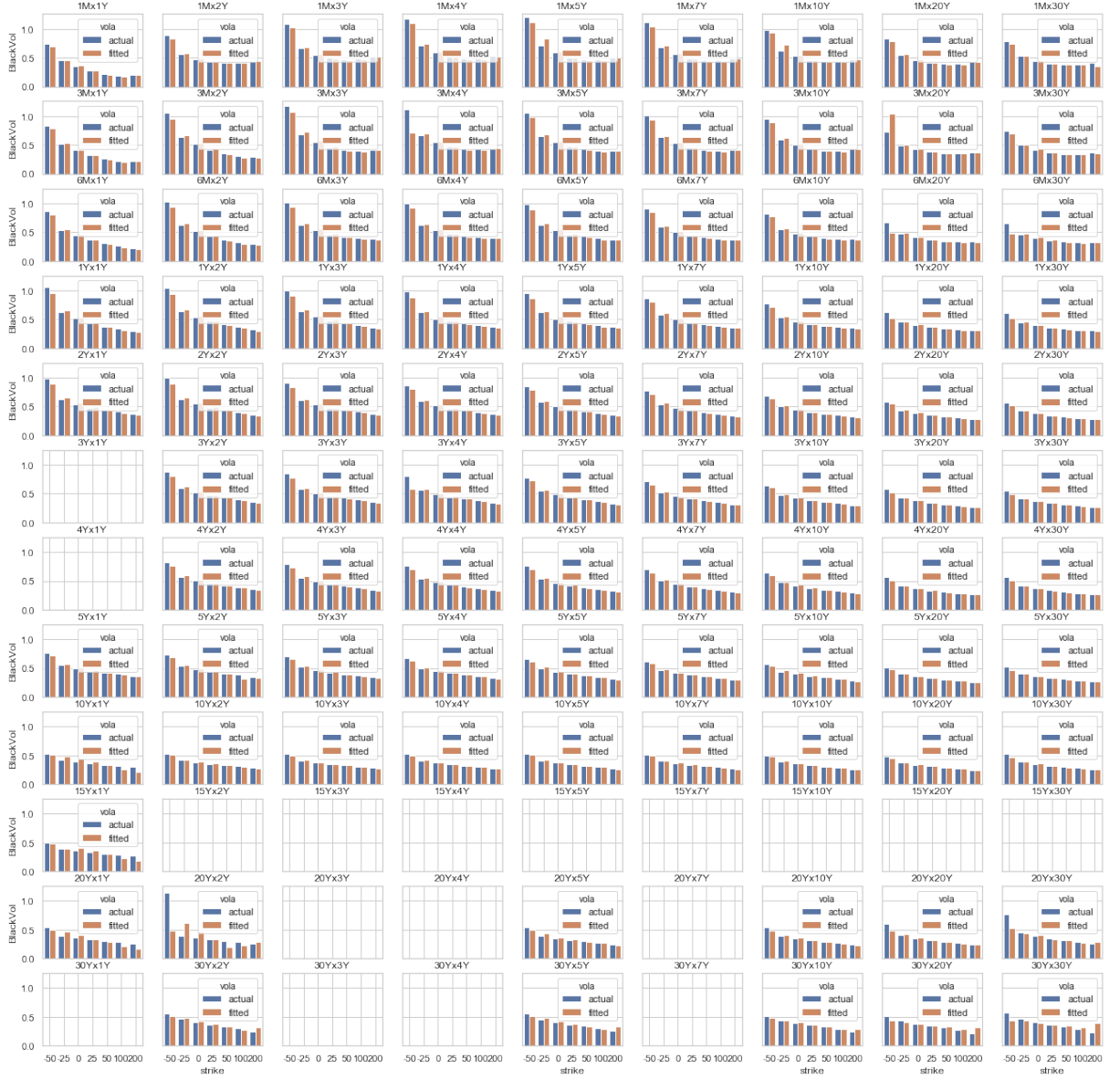


Figure 9: Leave one out approach performed for each strike of each smile on USD Black volatility cube. The bars on the left are actual values, the bars on the right are fitted values.

Expiry	Tenor	α	ν	ρ	Loss	Expiry	Tenor	α	ν	ρ	Loss
1M	1Y	0.035	1.138	-0.783	0.000	6M	10Y	0.043	0.778	-0.426	0.001
1M	2Y	0.046	0.989	-0.190	0.001	6M	20Y	0.036	0.652	-0.409	0.000
1M	3Y	0.047	1.102	-0.218	0.001	6M	30Y	0.034	0.612	-0.438	0.000
1M	4Y	0.047	1.119	-0.298	0.001	1Y	1Y	0.053	0.926	-0.761	0.003
1M	5Y	0.045	1.146	-0.367	0.002	1Y	2Y	0.058	0.826	-0.760	0.003
1M	7Y	0.042	1.049	-0.350	0.001	1Y	3Y	0.058	0.776	-0.645	0.002
1M	10Y	0.038	0.935	-0.317	0.001	1Y	4Y	0.054	0.817	-0.564	0.003
1M	20Y	0.030	0.781	-0.269	0.000	1Y	5Y	0.055	0.844	-0.587	0.004
1M	30Y	0.047	1.210	-0.665	0.013	1Y	7Y	0.053	0.796	-0.562	0.001
3M	1Y	0.042	1.112	-0.807	0.001	1Y	10Y	0.047	0.722	-0.495	0.001
3M	2Y	0.056	1.272	-0.781	0.003	1Y	20Y	0.041	0.607	-0.491	0.000
3M	3Y	0.049	1.253	-0.533	0.003	1Y	30Y	0.039	0.571	-0.492	0.000
3M	4Y	0.047	1.087	-0.431	0.002	2Y	1Y	0.056	0.655	-0.605	0.002
3M	5Y	0.050	1.047	-0.543	0.002	2Y	2Y	0.058	0.710	-0.667	0.002
3M	7Y	0.045	1.002	-0.482	0.002	2Y	3Y	0.057	0.646	-0.567	0.001
3M	10Y	0.038	0.936	-0.386	0.001	2Y	4Y	0.057	0.643	-0.573	0.001
3M	20Y	0.032	0.700	-0.313	0.000	2Y	5Y	0.058	0.684	-0.635	0.001
3M	30Y	0.028	0.693	-0.380	0.000	2Y	7Y	0.056	0.662	-0.616	0.001
6M	1Y	0.046	0.848	-0.841	0.001	2Y	10Y	0.053	0.633	-0.589	0.001
6M	2Y	0.057	1.038	-0.785	0.003	2Y	20Y	0.046	0.619	-0.587	0.001
6M	3Y	0.053	0.904	-0.542	0.002	2Y	30Y	0.043	0.578	-0.561	0.001
6M	4Y	0.050	0.869	-0.465	0.001	3Y	2Y	0.058	0.582	-0.652	0.001
6M	5Y	0.054	0.889	-0.570	0.002	3Y	3Y	0.058	0.608	-0.616	0.001
6M	7Y	0.049	0.846	-0.516	0.001	3Y	4Y	0.057	0.608	-0.621	0.001

Table 3: USD Libor calibration results of Black volatility cube. Part 1

mentioned techniques are not applied in here.

Subsequently, it is interesting to plot the ATM fitted volatility surface as figure 10. Along with the fitted surface, it is interesting to see how the actual at the money volatility surface looks like as figure 11.

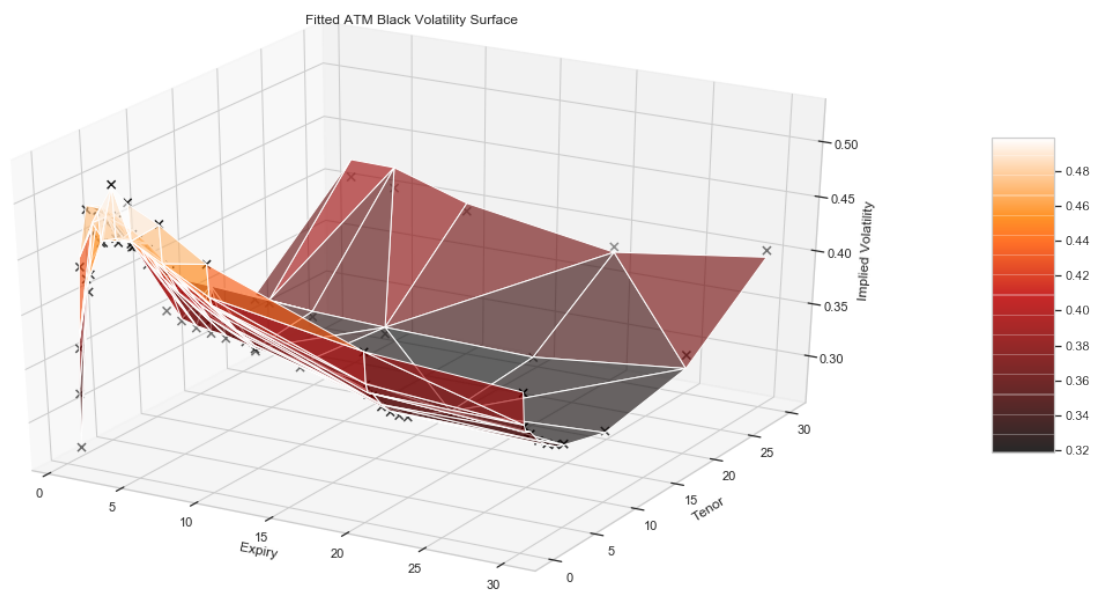


Figure 10: USD Black fitted volatility ATM surface. The points represent the actual volatility quotes.

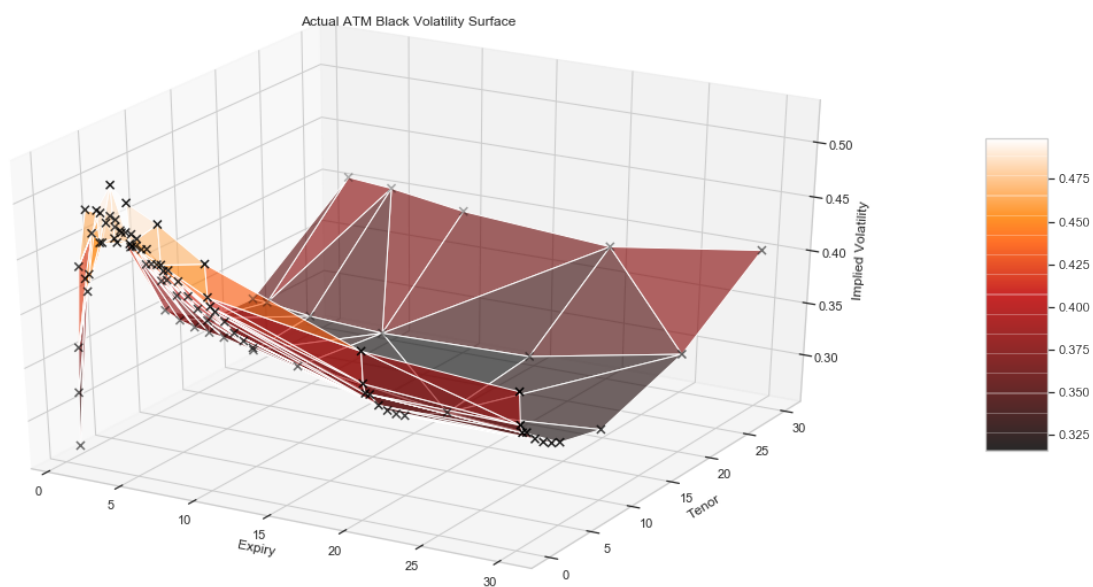


Figure 11: USD Black actual volatility ATM surface. The points represent the actual volatility quotes.

Expiry	Tenor	α	ν	ρ	Loss	Expiry	Tenor	α	ν	ρ	Loss
3Y	5Y	0.058	0.641	-0.667	0.001	10Y	1Y	0.049	0.399	-0.535	0.000
3Y	7Y	0.057	0.620	-0.651	0.001	10Y	2Y	0.049	0.417	-0.580	0.000
3Y	10Y	0.054	0.598	-0.629	0.000	10Y	3Y	0.048	0.428	-0.584	0.000
3Y	20Y	0.046	0.665	-0.610	0.001	10Y	4Y	0.048	0.431	-0.605	0.000
3Y	30Y	0.043	0.564	-0.570	0.001	10Y	5Y	0.049	0.478	-0.622	0.000
4Y	2Y	0.058	0.590	-0.637	0.001	10Y	7Y	0.048	0.461	-0.631	0.000
4Y	3Y	0.057	0.619	-0.602	0.001	10Y	10Y	0.049	0.457	-0.652	0.000
4Y	4Y	0.056	0.627	-0.613	0.001	10Y	20Y	0.047	0.521	-0.645	0.000
4Y	5Y	0.056	0.725	-0.650	0.001	10Y	30Y	0.043	0.536	-0.587	0.002
4Y	7Y	0.055	0.706	-0.647	0.001	15Y	1Y	0.178	3.466	-0.907	0.012
4Y	10Y	0.054	0.704	-0.647	0.001	20Y	1Y	0.045	0.473	-0.628	0.001
4Y	20Y	0.046	0.644	-0.612	0.001	20Y	2Y	0.049	2.137	-0.783	0.138
4Y	30Y	0.041	0.617	-0.580	0.002	20Y	5Y	0.046	0.547	-0.671	0.001
5Y	1Y	0.057	0.535	-0.539	0.001	20Y	10Y	0.047	0.602	-0.691	0.001
5Y	2Y	0.056	0.521	-0.610	0.001	20Y	20Y	0.042	0.651	-0.618	0.005
5Y	3Y	0.055	0.522	-0.587	0.000	20Y	30Y	0.033	0.735	-0.570	0.025
5Y	4Y	0.054	0.501	-0.591	0.000	30Y	2Y	0.055	0.612	-0.763	0.001
5Y	5Y	0.054	0.525	-0.623	0.000	30Y	5Y	0.058	0.628	-0.734	0.001
5Y	7Y	0.053	0.502	-0.630	0.000	30Y	10Y	0.063	0.578	-0.760	0.001
5Y	10Y	0.053	0.493	-0.649	0.000	30Y	20Y	0.055	0.452	-0.652	0.006
5Y	20Y	0.048	0.546	-0.624	0.000	30Y	30Y	0.049	0.542	-0.556	0.016
5Y	30Y	0.045	0.514	-0.580	0.001						

Table 4: USD Libor calibration results of Black volatility cube. Part 2

Conclusion

In the scope of the thesis we have discussed the mathematical preliminaries required for introducing the concepts of money-market derivatives. Subsequently, Black 76 and Bachelier models were discussed with their extensions for pricing the swaptions. As a last theoretical part, the SABR model is introduced and the Hagan's approximation formulas for implied Black and normal volatilities are shown, as well as the impact of the SABR parameters on the volatility smile is measured. As the thesis aims to cover both Euribor and USD Libor swaptions, the shifted SABR model is also introduced.

The last chapter covers the practical applications of SABR in terms of fitting the volatility cube of the Euribor and USD Libor swaptions, as well as the calibration procedure. An extra step is performed for the Euribor volatility cube calibration, as the market quotes are in terms of normal volatility and for negative rates and strikes, while the Hagan's approximation formulas are suited for the positive ones. The extra step involves shifting the normal volatility cube to a shifted Black volatility quotes.

The results of the calibration are then evaluated by leaving one strike out and estimating the fitted and actual volatility quotes differences. The evaluation shows an acceptable fit and can possibly be used for pricing actual swaptions using the calibrated volatility cube. Some larger errors appear for the strikes that are on the right and left edges of the actual strike intervals, which can be explained by the fact that they are on the edges of the strike interval and the model is naturally expressing worse fit for those areas. Another reason for non perfect fits is the market liquidity and the possible bid-ask spreads that occur for the illiquid parts of the volatility cube, as well as some market data noise. Summing up the results of the thesis, we can conclude that the SABR model is a powerful tool for fitting the market volatility cube and can, indeed, be used for pricing swaptions in practice.

References

- [1] Patrick S. Hagan, Deep Kumar, Andrew S. Lesniewski and Diana E. Woodward (2002) *Managing Smile Risk*, Wilmott
- [2] Damiano Brigo, Fabio Mercurio (2001) *Pricing of Commodity Contracts, Interest Rate Models Theory and Practice*, *Journal of Financial Economics*, 167-179 pp.
- [3] John Hull (2008) *Options, Futures And Other Derivatives*, 7th edition
- [4] E.H. Soerensen and T.F. Bollier (1994) *Pricing Swap Default Risk. Financial Analysts Journal*

Interest Rate Models Theory and Practice 2001 edition by Damiano Brigo, Fabio Mercurio, Pricing of Commodity Contracts, *Journal of Financial Economics* 3 (1976) 167-179 pp
- [5] Schumacher, J.M. (2010). *Financial Models. Tilburg University, the Netherlands*
- [6] Kazuhiro Iwasawa. *Analytic formula for the european normal black scholes formula.*

Appendix

```
1
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 import math
6 import datetime
7 import scipy as sp
8 from scipy.stats import norm
9 from scipy.optimize import minimize, minimize_scalar, fsolve #, rosen,
    rosen_der
10 import seaborn as sns
11 from scipy.interpolate import interp2d
12 from matplotlib import cm
13 from mpl_toolkits import mplot3d
14 sns.set(style="whitegrid")
15 import matplotlib.pyplot as plt
16 from mpl_toolkits.mplot3d import Axes3D
17
18
19 def standard_time(t):
20     # is used for the plot titles only.
21     if t<1:
22         if np.abs(t*365 - 30) <5:
23             result = '1M'
24         elif np.abs(t*365 - 90) <10:
25             result = '3M'
26         elif np.abs(t*365 - 180) <15:
27             result = '6M'
28         elif np.abs(t*365 - 270) <20:
29             result = '9M'
30         else:
31             result = t
32     else:
33         result = '{}Y'.format(int(t))
34     return result
```

```

35
36
37 #Black76 For EUR
38 def Annuity(T_0, T_mn, tenordelta):
39     return tenordelta * np.sum(DF(np.arange(T_0, T_0 + T_mn, tenordelta)
40     + tenordelta, euros))
41
42 def SwaptionB76(t, T_0, T_mn, r_k, sigma, S_t, tenordelta):
43     T = T_0 - t
44     d1 = (np.log(S_t / r_k) + 1/2*sigma**2*T) / (sigma*np.sqrt(T))
45     d2 = d1 - sigma*np.sqrt(T)
46
47     A_0_mn = Annuity(T_0, T_mn, tenordelta)
48
49     return A_0_mn * (S_t * norm.cdf(d1) - r_k * norm.cdf(d2))
50
51
52 #Bachelier
53 def SwaptionBachelier(t, T_0, T_mn, r_k, sigma, S_t, tenordelta):
54     T = T_0 - t
55     d = (S_t - r_k) / (sigma * np.sqrt(T))
56
57     A_0_mn = Annuity(T_0, T_mn, tenordelta)
58
59     return A_0_mn * sigma * np.sqrt(T) * (d*norm.cdf(d) + norm.pdf(d))
60
61
62 def sigma_B_general(K, f, alpha, nu, rho, T):
63     def x(z):
64         return np.log((np.sqrt(1-2*rho*z + z**2) + z - rho)/(1 - rho))
65     z = (nu/alpha) * (f*K)**((1-beta)/2)*np.log(f/K)
66     sigma = (alpha/((f*K)**((1-beta)/2) * (1 + ((1-beta)**2/24)* (np.log
67     (f/K))**2 + ((1-beta)**4/1920)* (np.log(f/K))**4 ) )) * (z/x(z)) * (
68     1 + ( ((1-beta)**2/24) * (alpha**2/(f*K)**(1-beta)) + 1/4* (rho*beta
69     *nu*alpha)/(f*K)**((1-beta)/2) + (2-3*rho**2)/24*nu**2)*T )
70     return sigma

```

```

69
70
71 def sigma_B_atm(f, alpha, nu, rho, T):
72     sigma = alpha/f**(1-beta)*(1 + (
73         (1-beta)**2/24*alpha**2/f**(2-2*beta) + 1/4*rho*beta*alpha*nu/f
74         **(1-beta) + (2-3*rho**2)/24*nu**2 )*T)
75     return sigma
76
77 def sigma_B(K, f, alpha, nu, rho, T):
78     if K==f:
79         return sigma_B_atm(f, alpha, nu, rho, T)
80     else:
81         return sigma_B_general(K, f, alpha, nu, rho, T)
82
83
84 def sigma_N_general(K, f, alpha, nu, rho, T):
85     def x(z):
86         return np.log((np.sqrt(1-2*rho*z + z**2) + z - rho)/(1 - rho))
87     z = (nu/alpha) * (f*K)**((1-beta)/2)*np.log(f/K)
88     sigma = alpha*(f*K)**(beta/2) * (1 + 1/24*np.log(f/K)**2 + 1/1920 *
89     np.log(f/K)**4 ) / (
90         1 + (1-beta)**2 * 1/24 * np.log(f/K)**2 + (1-beta)**4/1920 * np.
91         log(f/K)**4) * (z/x(z)) * (
92         1 + ( (-beta*(2-beta)*alpha**2)/(24*(f*K)**(1-beta)) + 1/4* (rho*
93         beta*nu*alpha)/(f*K)**((1-beta)/2) + (2-3*rho**2)/24*nu**2)*T )
94     return sigma
95
96
97 def sigma_N_atm(f, alpha, nu, rho, T):
98     sigma = alpha*f**beta * (1 + (
99         (-beta*(2-beta)*alpha**2)/(24*f**(2-2*beta)) + 1/4*rho*beta*
100         alpha*nu/f**(1-beta) + (2-3*rho**2)/24*nu**2 )*T)
101     return sigma
102
103 def sigma_N(K, f, alpha, nu, rho, T):

```

```

101     if K==f:
102         return sigma_N_atm(f, alpha, nu, rho, T)
103     else:
104         return sigma_N_general(K, f, alpha, nu, rho, T)
105
106
107 # Read the data
108 vols = pd.read_csv('VCUB_20200204.csv', sep=';', decimal=',')
109 eur = pd.read_csv('EUR6M.csv')
110 eur.Maturity_Date = pd.to_datetime(eur.Maturity_Date)
111 eurois = pd.read_csv('EUROIS.csv')
112 eurois.Maturity_Date = pd.to_datetime(eurois.Maturity_Date)
113 eurois['tenor'] = eurois.Maturity_Date.apply(lambda x: (x - eurois.
    Maturity_Date[0]).days/365)
114 eur['tenor'] = eur.Maturity_Date.apply(lambda x: (x - eur.Maturity_Date
    [0]).days/365)
115 eur['spot'] = eur.apply(lambda x: x['DiscFactor']*(-1/x['tenor'] if x['
    tenor']>0 else 0) -1, axis=1)
116 eurois['spot'] = eurois.apply(lambda x: x['DiscFactor']*(-1/x['tenor']
    if x['tenor']>0 else 0) -1, axis=1)
117
118
119 def spot(tenor, curve):
120     return np.interp(tenor, xp=curve['tenor'], fp=curve['spot'])
121
122
123 def DF(tenor, curve):
124     return 1/(1 + spot(tenor, curve))*tenor
125
126
127 def F(T1, T2, curve):
128     # Forward rate
129     # this is actually F(t, T1, T2), t is 2020-02-04
130     return 1/(T2 - T1)*(DF(T1, curve)/DF(T2, curve) - 1)
131
132
133 def S(T1, T2, curve, step=0.5):

```

```

134     # Forward-Swap rate
135     # this is actually S(t, T1, T2), t is 2020-02-04
136
137     dots = np.arange(T1,T2,step) + step
138     denominator = np.sum(DF(dots, curve)*step)
139     return (DF(T1, curve) - DF(T2, curve)) / denominator
140
141
142 # shifting the inputs
143 def implied_black_loss(sigma, t, T_0, T_mn, r_k, S_t, tenordelta, price)
144 :
145     #sigma = sigma_array[0]
146     return (SwaptionB76(t, T_0, T_mn, r_k+0.03, sigma, S_t+0.03,
147         tenordelta) - price )
148
149
150 vols['BlackVol'] = np.nan
151 vols['Price'] = np.nan
152 vols['B76Price'] = np.nan
153 vols['Loss'] = np.nan
154 implied_volas = dict()
155 loss_implied = dict()
156 for i, v in vols.iterrows():
157
158     expiry = v['Term']
159     tenor = v['Tenor']
160     sigma = v['NVol']/10000
161     strike = v['Strike']/10000
162     price = SwaptionBachelier(0, expiry, tenor, S(expiry, expiry + tenor
163         , eur) + strike, sigma, S(expiry, expiry + tenor, eur), 0.5)
164     solution = fsolve(implied_black_loss, 0.5, args=(0, expiry, tenor,
165         S(expiry, expiry + tenor, eur) + strike, S(expiry, expiry + tenor,
166         eur), 0.5, price)
167         , xtol=0.000001)
168
169     vols.loc[((vols.Term == expiry) & (vols.Tenor == tenor) & (vols.
170         Strike == v['Strike'])), 'B76Price'] = SwaptionB76(0, expiry, tenor,

```



```

    strike+S(expiry, expiry + tenor, eur)+0.03, solution, S(expiry,
    expiry + tenor, eur)+0.03, 0.5)
165     vols.loc[((vols.Term == expiry) & (vols.Tenor == tenor) & (vols.
        Strike == v['Strike']))], 'BlackVol'] = solution
166     vols.loc[((vols.Term == expiry) & (vols.Tenor == tenor) & (vols.
        Strike == v['Strike']))], 'Price'] = price
167 vols['Loss'] = vols['Price'] / vols['B76Price'] -1
168
169
170 # # Fitting SABR
171 def smile_loss(x, expiry, tenor, vols, sigma_function, volatility_name,
    s=0, curve=eur):
172     alpha = x[0]
173     nu = x[1]
174     rho = x[2]
175     loss = 0
176
177     atm = S(expiry, expiry + tenor, curve)
178
179     smile = vols[(vols.Term == expiry) & (vols.Tenor == tenor)][['Strike
        ', volatility_name]]
180     result = np.array([sigma_function(s+k/10000+atm, s+atm, alpha, nu,
        rho, tenor) for k in smile['Strike'].values])
181
182     loss += sum((result - smile[volatility_name])**2)
183     return loss
184
185 #fix some params
186 beta = 1
187
188 # filter the volatilities for which the price is too small
189 vols_filtered = vols[vols.Price > 1/ 10**9].copy()
190
191
192 START = datetime.datetime.now()
193 x0 = [0.6, 0.4, -0.2]
194 expiry_list = []

```

```

195 tenor_list = []
196 alpha_list = []
197 nu_list = []
198 rho_list = []
199 loss_list = []
200 for expiry in vols_filtered.Term.unique():
201     for tenor in vols_filtered.Tenor.unique():
202         solution = minimize(fun=smile_loss, x0=x0 , args=(
203             expiry, tenor, vols_filtered, sigma_B, 'BlackVol', 0.03, eur
204             ), bounds = (
205                 (0.001, None), (0.001, None), (-0.999, 0.999) ), method="SLSQP", tol
206                 =1e-20)
207         expiry_list.append(expiry)
208         tenor_list.append(tenor)
209         alpha_list.append(solution.x[0])
210         nu_list.append(solution.x[1])
211         rho_list.append(solution.x[2])
212         loss_list.append(solution.fun)
213 calibration_result = pd.DataFrame({'expiry': expiry_list
214                                     , 'tenor': tenor_list
215                                     , 'alpha': alpha_list
216                                     , 'nu': nu_list
217                                     , 'rho': rho_list
218                                     , 'loss': loss_list})
219 print('duration: ', datetime.datetime.now() - START)
220
221 calibration_result.to_csv('EUR_Calibration.csv')
222
223 [k for k in vols_filtered[(vols_filtered.Term == expiry) & (
224     vols_filtered.Tenor == tenor)]['Strike']/10000]
225
226 fig, ax = plt.subplots(len(calibration_result.expiry.unique()), len(
227     calibration_result.tenor.unique())
228                         , sharex=True, sharey=True, figsize=(15,15))

```

```

228 for i, expiry in enumerate(calibration_result.expiry.unique()):
229     for j, tenor in enumerate(calibration_result.tenor.unique()):
230
231         params = calibration_result[(calibration_result.expiry == expiry
232 ) & (calibration_result.tenor == tenor)]
233
234         alpha = params.alpha.values[0]
235         nu = params.nu.values[0]
236         rho = params.rho.values[0]
237         atm = S(expiry, expiry+tenor, eur)
238         sns.lineplot(x=[k for k in vols[(vols.Term == expiry) & (vols.
239 Tenor == tenor)]['Strike']]
240
241             , y=[sigma_B(0.03 + atm + k, 0.03 + atm, alpha, nu,
242 rho, tenor) for k in vols[(vols.Term == expiry) & (vols.Tenor ==
243 tenor)]['Strike']]/10000]
244
245             , ax=ax[i, j])
246         sns.scatterplot(x=[k for k in vols_filtered[(vols_filtered.Term
247 == expiry) & (vols_filtered.Tenor == tenor)]['Strike']]
248
249             , y=[vol for vol in vols_filtered[(vols_filtered.
250 Term == expiry) & (vols_filtered.Tenor == tenor)]['BlackVol']]
251
252             , ax= ax[i, j], color='red')
253         ax[i, j].set_title('{x}'.format(standard_time(expiry),
254 standard_time(tenor)))
255 fig.savefig('EUR_smiles.png')
256
257 # # Leave one out
258 idx = vols_filtered[(vols_filtered.Term==2) & (vols_filtered.Tenor==5) &
259 (vols_filtered.Strike==-50)].index
260 START = datetime.datetime.now()
261 x0 = [0.6, 0.4, -0.2]
262 expiry_list = []
263 tenor_list = []
264 alpha_list = []
265 nu_list = []
266 rho_list = []
267 loss_list = []
268 strike_list = []
269 for expiry in vols_filtered.Term.unique():

```

```

257     for tenor in vols_filtered.Tenor.unique():
258         for strk in vols_filtered[(vols_filtered.Term==expiry) & (
vols_filtered.Tenor==tenor)].Strike:
259             # index to leave out
260             idx = vols_filtered[(vols_filtered.Term==expiry) & (
vols_filtered.Tenor==tenor) & (vols_filtered.Strike==strk)].index
261
262             solution = minimize(fun=smile_loss, x0=x0 , args=(
263                 expiry, tenor, vols_filtered.drop(index=idx), sigma_B, '
BlackVol', 0.03, eur), bounds = (
264                 (0.001, None), (0.001, None), (-0.999, 0.999) ), method="SLSQP",
tol=1e-20)
265             expiry_list.append(expiry)
266             tenor_list.append(tenor)
267             strike_list.append(strk)
268             alpha_list.append(solution.x[0])
269             nu_list.append(solution.x[1])
270             rho_list.append(solution.x[2])
271             loss_list.append(solution.fun)
272 leave_one_out_result = pd.DataFrame({'expiry': expiry_list
273                                     , 'tenor': tenor_list
274                                     , 'strike': strike_list
275                                     , 'alpha': alpha_list
276                                     , 'nu': nu_list
277                                     , 'rho': rho_list
278                                     , 'loss': loss_list})
279 print('duration: ', datetime.datetime.now() - START)
280
281
282 # calculate errors for each left one out
283 leave_one_out_result['sigma_hat'] = leave_one_out_result.apply(lambda x:
284     sigma_B(
285         0.03 + S(x['expiry'], x['expiry']+x['tenor'], eur) + x['strike'
286         ]/10000, 0.03 + S(x['expiry'], x['expiry']+x['tenor'], eur), x['alpha
287         '], x['nu'], x['rho'], x['tenor']))
288         , axis=1)

```

```

287
288 leave_one_out_result = pd.merge(leave_one_out_result, vols_filtered[['
    Term', 'Tenor', 'Strike', 'BlackVol']]
289     , left_on = ['expiry', 'tenor', 'strike'], right_on=['Term', '
    Tenor', 'Strike']
290     , how='left').drop(columns=['Term', 'Tenor', 'Strike'])
291
292
293 actual_df = leave_one_out_result.drop(columns='sigma_hat')
294 fitted_df = leave_one_out_result.drop(columns='BlackVol').rename(columns
    ={'sigma_hat': 'BlackVol'})
295 actual_df['vola'] = 'actual'
296 fitted_df['vola'] = 'fitted'
297
298 data_to_plot = pd.concat([actual_df, fitted_df], ignore_index=True)
299
300
301 sns.barplot(x='strike', y='BlackVol', hue='vola', data=data_to_plot[(
    data_to_plot.expiry == 5) & (data_to_plot.tenor == 5)])
302
303
304 fig, ax = plt.subplots(len(data_to_plot.expiry.unique()), len(
    data_to_plot.tenor.unique())
305                        , sharex=True, sharey=True, figsize=(22,22))
306
307
308 for i, expiry in enumerate(calibration_result.expiry.unique()):
309     for j, tenor in enumerate(calibration_result.tenor.unique()):
310         sns.barplot(x='strike', y='BlackVol', hue='vola', ax=ax[i, j]
311                    , data=data_to_plot[(data_to_plot.expiry == expiry)
312                    & (data_to_plot.tenor == tenor)])
313         ax[i, j].set_title('{}x{}'.format(standard_time(expiry),
314         standard_time(tenor)))
315         if expiry<30:
316             ax[i, j].set_xlabel('')
317         if tenor>2:
318             ax[i, j].set_ylabel('')

```

```

317 fig.savefig('EUR_barplot.png')
318 plt.show()
319
320
321 # surface plots
322 data3 = data_to_plot[(data_to_plot.strike==0) & (data_to_plot.vola=='
    fitted')]
323 zdata = data3['BlackVol']
324 xdata = data3['tenor']
325 ydata = data3['expiry']
326 X_fitted = xdata.values
327 Y_fitted = ydata.values
328 Z_fitted = zdata.values
329
330 data3 = data_to_plot[(data_to_plot.strike==0) & (data_to_plot.vola=='
    actual')]
331 zdata = data3['BlackVol']
332 xdata = data3['tenor']
333 ydata = data3['expiry']
334 X_actual = xdata.values
335 Y_actual = ydata.values
336 Z_actual = zdata.values
337
338
339 fig = plt.figure(figsize=(20, 10))
340 ax = fig.add_subplot(111, projection='3d')
341 ax.set_xlabel('Expiry')
342 ax.set_ylabel('Tenor')
343 ax.set_zlabel('Implied Volatility')
344 ax.set_title('Actual ATM Black Volatility Surface')
345
346 ax.scatter3D(X_actual, Y_actual, Z_actual+0.0001, c=20*np.ones(len(
    Z_actual)), cmap='gist_heat', s=50, marker='x')
347
348 surf = ax.plot_trisurf(X_actual, Y_actual, Z_actual, cmap='gist_heat',
    alpha=0.6)
349 fig.colorbar(surf, shrink=0.5, aspect=5, alpha=0.4)

```

```

350 fig.savefig('EUR_Surface_actual.png')
351
352 fig = plt.figure(figsize=(20, 10))
353 ax = fig.add_subplot(111, projection='3d')
354 ax.set_xlabel('Expiry')
355 ax.set_ylabel('Tenor')
356 ax.set_zlabel('Implied Volatility')
357 ax.set_title('Fitted ATM Black Volatility Surface')
358 surf = ax.plot_trisurf(X_fitted, Y_fitted, Z_fitted, cmap='gist_heat',
    alpha=0.6)
359
360
361 ax.scatter3D(X_actual, Y_actual, Z_actual+0.0001, c=20*np.ones(len(
    Z_actual)), cmap='gist_heat', s=50, marker='x')
362 fig.colorbar(surf, shrink=0.5, aspect=5)
363 fig.savefig('EUR_Surface_fitted.png')
364
365
366
367
368
369 ##### USD LIBOR CASE
370 def Annuity(T_0, T_mn, tenordelta):
371     return tenordelta * np.sum(DF(np.arange(T_0, T_0 + T_mn, tenordelta)
    + tenordelta, usdois))
372
373
374
375 usd = pd.read_csv('USD3M.csv', sep=';', decimal=',')
376 usd.Maturity_Date = pd.to_datetime(usd.Maturity_Date)
377 usdois = pd.read_csv('USD0IS.csv', sep=';', decimal=',')
378 usdois.Maturity_Date = pd.to_datetime(usdois.Maturity_Date)
379 usdois['tenor'] = usdois.Maturity_Date.apply(lambda x: (x - usdois.
    Maturity_Date[0]).days/365)
380 usd['tenor'] = usd.Maturity_Date.apply(lambda x: (x - usd.Maturity_Date
    [0]).days/365)
381 usd['spot'] = usd.apply(lambda x: x['DiscFactor']*(-1/x['tenor'] if x['

```

```

        tenor']>0 else 0) -1, axis=1)
382 usdois['spot'] = usdois.apply(lambda x: x['DiscFactor']**(-1/x['tenor']
        if x['tenor']>0 else 0) -1, axis=1)
383 vols = pd.read_csv('VCUB_USD_20200204.csv', sep=';', decimal=',')
384 vols = pd.merge(vols, vols[['Term', 'Tenor', 'BlackVol']][vols.Strike
        ==0].rename(
385     columns={'BlackVol': 'ATM'}), on=['Term', 'Tenor'], how='left')
386 vols.loc[(vols.Strike==0), 'ATM'] = 0
387 vols['BlackVol'] = vols['BlackVol'] + vols['ATM']
388 vols = vols.drop(columns='ATM')
389 vols['BlackVol'] = vols['BlackVol']/100
390 vols = vols.dropna()
391 vols = vols.sort_values(by='Strike')
392
393
394 # # Fitting SABR
395
396 def smile_loss(x, expiry, tenor, vols, sigma_function, volatility_name,
        s=0, curve=usd):
397     alpha = x[0]
398     nu = x[1]
399     rho = x[2]
400     loss = 0
401
402     atm = S(expiry, expiry + tenor, curve)
403
404     smile = vols[(vols.Term == expiry) & (vols.Tenor == tenor)][['Strike',
        volatility_name]]
405     result = np.array([sigma_function(s+k/10000+atm, s+atm, alpha, nu,
        rho, tenor) for k in smile['Strike'].values])
406
407     loss += sum((result - smile[volatility_name])**2)
408     return loss
409
410
411
412 #fix some params

```



```

413 beta = 0.5
414
415 START = datetime.datetime.now()
416 x0 = [0.6, 0.4, -0.2]
417 expiry_list = []
418 tenor_list = []
419 alpha_list = []
420 nu_list = []
421 rho_list = []
422 loss_list = []
423 for i, v in vols[['Term', 'Tenor']].drop_duplicates().iterrows():
424     expiry = v['Term']
425     tenor = v['Tenor']
426     solution = minimize(fun=smile_loss, x0=x0, args=(
427         expiry, tenor, vols, sigma_B, 'BlackVol', 0, usd), bounds = (
428 (0.001, None), (0.001, None), (-0.999, 0.999)), method="SLSQP", tol=1e
429 -20)
429     expiry_list.append(expiry)
430     tenor_list.append(tenor)
431     alpha_list.append(solution.x[0])
432     nu_list.append(solution.x[1])
433     rho_list.append(solution.x[2])
434     loss_list.append(solution.fun)
435 calibration_result = pd.DataFrame({'expiry': expiry_list
436     , 'tenor': tenor_list
437     , 'alpha': alpha_list
438     , 'nu': nu_list
439     , 'rho': rho_list
440     , 'loss': loss_list})
441 print('duration: ', datetime.datetime.now() - START)
442
443 calibration_result = calibration_result.sort_values(by=['expiry', 'tenor
444     '])
445 calibration_result.to_csv('USD_calibration.csv')
446
447

```

```

448 fig, ax = plt.subplots(len(calibration_result.expiry.unique()), len(
    calibration_result.tenor.unique())
449                          , sharex=True, sharey=True, figsize=(15,15))
450 for i, expiry in enumerate(calibration_result.expiry.unique()):
451     for j, tenor in enumerate(calibration_result.tenor.unique()):
452
453         try:
454             params = calibration_result[(calibration_result.expiry ==
455 expiry) & (calibration_result.tenor == tenor)]
456             alpha = params.alpha.values[0]
457             nu = params.nu.values[0]
458             rho = params.rho.values[0]
459             atm = S(expiry, expiry+tenor, usd)
460             sns.lineplot(x=[k for k in vols[(vols.Term == expiry) & (
461 vols.Tenor == tenor)]]['Strike']]
462                          , y=[sigma_B(atm + k, atm, alpha, nu, rho,
463 tenor) for k in vols[(vols.Term == expiry) & (vols.Tenor == tenor)]]['
464 Strike']/10000]
465                          , ax=ax[i, j])
466             sns.scatterplot(x=[k for k in vols[(vols.Term == expiry) & (vols
467 .Tenor == tenor)]]['Strike']]
468                          , y=[vol for vol in vols[(vols.Term == expiry
469 ) & (vols.Tenor == tenor)]]['BlackVol']]
470                          , ax= ax[i, j], color='red')
471             ax[i, j].set_title('{x}'.format(standard_time(expiry),
472 standard_time(tenor)))
473         except:
474             ax[i, j].set_title('{x}'.format(standard_time(expiry),
475 standard_time(tenor)))
476         pass
477 fig.savefig('USD_smiles.png')
478
479 # # Leave one out
480 START = datetime.datetime.now()
481 x0 = [0.6, 0.4, -0.2]
482 expiry_list = []
483 tenor_list = []

```

```

476 alpha_list = []
477 nu_list = []
478 rho_list = []
479 loss_list = []
480 strike_list = []
481 for expiry in vols.Term.unique():
482     for tenor in vols.Tenor.unique():
483         for strk in vols[(vols.Term==expiry) & (vols.Tenor==tenor)].
Strike:
484             # index to leave out
485
486             idx = vols[(vols.Term==expiry) & (vols.Tenor==tenor) & (vols
.Strike==strk)].index
487
488             solution = minimize(fun=smile_loss, x0=x0 , args=(
489                 expiry, tenor, vols.drop(index=idx), sigma_B, 'BlackVol'
, 0, usd), bounds = (
490                 (0.001, None), (0.001, None), (-0.999, 0.999) ), method="SLSQP",
tol=1e-20)
491             expiry_list.append(expiry)
492             tenor_list.append(tenor)
493             strike_list.append(strk)
494             alpha_list.append(solution.x[0])
495             nu_list.append(solution.x[1])
496             rho_list.append(solution.x[2])
497             loss_list.append(solution.fun)
498 leave_one_out_result = pd.DataFrame({'expiry': expiry_list
499                                     , 'tenor': tenor_list
500                                     , 'strike': strike_list
501                                     , 'alpha': alpha_list
502                                     , 'nu': nu_list
503                                     , 'rho': rho_list
504                                     , 'loss': loss_list})
505 print('duration: ', datetime.datetime.now() - START)
506
507
508 # calculate errors for each left one out

```

```

509 leave_one_out_result['sigma_hat'] = leave_one_out_result.apply(lambda x:
    sigma_B(
510     S(x['expiry'], x['expiry']+x['tenor'], usd) + x['strike']/10000,
    S(x['expiry'], x['expiry']+x['tenor'], usd), x['alpha'], x['nu'], x['
    rho'], x['tenor'])
511     , axis=1)
512
513
514 leave_one_out_result = pd.merge(leave_one_out_result, vols[['Term', '
    Tenor', 'Strike', 'BlackVol']])
515     , left_on = ['expiry', 'tenor', 'strike'], right_on=['Term', '
    Tenor', 'Strike']
516     , how='left').drop(columns=['Term', 'Tenor', 'Strike'])
517
518
519 actual_df = leave_one_out_result.drop(columns='sigma_hat')
520 fitted_df = leave_one_out_result.drop(columns='BlackVol').rename(columns
    ={'sigma_hat': 'BlackVol'})
521 actual_df['vola'] = 'actual'
522 fitted_df['vola'] = 'fitted'
523 data_to_plot = pd.concat([actual_df, fitted_df], ignore_index=True)
524 data_to_plot = data_to_plot.sort_values(by=['expiry', 'tenor'])
525 sns.barplot(x='strike', y='BlackVol', hue='vola', data=data_to_plot[(
    data_to_plot.expiry == 5) & (data_to_plot.tenor == 5)])
526
527 fig, ax = plt.subplots(len(data_to_plot.expiry.unique()), len(
    data_to_plot.tenor.unique())
528     , sharex=True, sharey=True, figsize=(22,22))
529
530
531 for i, expiry in enumerate(calibration_result.expiry.unique()):
532     for j, tenor in enumerate(calibration_result.tenor.unique()):
533         try:
534             sns.barplot(x='strike', y='BlackVol', hue='vola', ax=ax[i, j
    ]
535                 , data=data_to_plot[(data_to_plot.expiry ==
    expiry) & (data_to_plot.tenor == tenor)])

```

```

536         ax[i, j].set_title('{}x{}'.format(standard_time(expiry),
standard_time(tenor)))
537     except:
538         ax[i, j].set_title('{}x{}'.format(standard_time(expiry),
standard_time(tenor)))
539     pass
540
541     if expiry<30:
542         ax[i, j].set_xlabel('')
543     if tenor>1:
544         ax[i, j].set_ylabel('')
545
546 plt.show()
547 fig.savefig('USD_barplot.png')
548
549 # surface plots
550 data3 = data_to_plot[(data_to_plot.strike==0) & (data_to_plot.vola=='
fitted')]
551 zdata = data3['BlackVol']
552 xdata = data3['tenor']
553 ydata = data3['expiry']
554 X_fitted = xdata.values
555 Y_fitted = ydata.values
556 Z_fitted = zdata.values
557
558 data3 = data_to_plot[(data_to_plot.strike==0) & (data_to_plot.vola=='
actual')]
559 zdata = data3['BlackVol']
560 xdata = data3['tenor']
561 ydata = data3['expiry']
562 X_actual = xdata.values
563 Y_actual = ydata.values
564 Z_actual = zdata.values
565
566
567 fig = plt.figure(figsize=(20, 10))
568 ax = fig.add_subplot(111, projection='3d')

```

```

569 ax.set_xlabel('Expiry')
570 ax.set_ylabel('Tenor')
571 ax.set_zlabel('Implied Volatility')
572 ax.set_title('Actual ATM Black Volatility Surface')
573
574 ax.scatter3D(X_actual, Y_actual, Z_actual+0.0001, c=20*np.ones(len(
    Z_actual)), cmap='gist_heat', s=50, marker='x')
575
576 surf = ax.plot_trisurf(X_actual, Y_actual, Z_actual, cmap='gist_heat',
    alpha=0.6)
577 fig.colorbar(surf, shrink=0.5, aspect=5, alpha=0.4)
578 fig.savefig('USD_surface_actual.png')
579
580
581 fig = plt.figure(figsize=(20, 10))
582 ax = fig.add_subplot(111, projection='3d')
583 ax.set_xlabel('Expiry')
584 ax.set_ylabel('Tenor')
585 ax.set_zlabel('Implied Volatility')
586 ax.set_title('Fitted ATM Black Volatility Surface')
587 surf = ax.plot_trisurf(X_fitted, Y_fitted, Z_fitted, cmap='gist_heat',
    alpha=0.6)
588
589
590 ax.scatter3D(X_actual, Y_actual, Z_actual+0.0001, c=20*np.ones(len(
    Z_actual)), cmap='gist_heat', s=50, marker='x')
591 fig.colorbar(surf, shrink=0.5, aspect=5)
592 fig.savefig('USD_surface_fitted.png')

```

Non-exclusive licence to reproduce thesis and make thesis public

I, Karen Danielyan,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,
Swaption Pricing with SABR model,
supervised by Raul Kangro.
2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.
4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Karen Danielyan

26/05/2020